The Expressive Power of Delay Operators in SCCS

Carol Critchlow
Prakash Panangaden

TR 89-998
May 1989

Department of Computer Science
Cornell University
Ithaca, NY 14853-7501

The Expressive Power of Delay Operators in SCCS

Carol Critchlow
Prakash Panangaden
Computer Science Department, Cornell University

April 30, 1989

Abstract

We investigate the relative expressive power of finite delay operators in SCCS. These were introduced by Milner and by Hennessy to study fairness properties of processes in the context of SCCS. We show that the context sensitive delay operator introduced by Hennessy is more expressive than the finite delay operator introduced by Milner. This result is closely related to recent results by Pananagden and Stark on the expresive power of fair merge in asynchronous dataflow (Kahn) networks. It indicates that the expressiveness results obtained there are not sensitive to the precise computational model since SCCS, unlike Kahn networks, is synchronous and permits expansion of recursively defined processes.

1 Introduction

Recent work by Panangaden and Stark [4] has established that there is a difference in the expressive power of indeterminate merge primitives in the context of static dataflow networks. More precisely, it was shown that the fair merge primitive could not be expressed in terms of primitives exhibiting unbounded indeterminacy. In the present paper we examine a related expressiveness situation in the context of Milner's Synchronous Calculus of Communicating Systems (SCCS) [2,3]. The point of the present investigation is to emphasize that the expressiveness results obtained earlier for static, asynchronous dataflow are, in some sense, not sensitive to the model of computation that is used in the analysis. SCCS is synchronous and permits unwinding of recursive process descriptions; two major differences from dataflow networks. The technical details of the proofs are quite different in the two cases though the phenomena are similar.

The primitives that we study in the present paper are two finite-delay operators introduced by Milner [2] and by Hennessy [1]. Milner's delay operator, written ε, will delay a process only finitely many steps; Hennessy's delay operator, written γ, will delay a process finitely *unless it has nothing to synchronize with.* Thus γ is a context-sensitive operator.

We first analyze the delay operators in a simplified process algebra used by Hennessy [1] and then discuss the situation in SCCS.

2 Delay Operators in a Simple Process Algebra

We begin by summarizing the language and the transition rules of Hennessy's system [1]. Let A be a set of actions, and define the set of terms over A by

$$t ::= \emptyset|\Omega|x|at|t_1 + t_2|\varepsilon t|\delta t|\gamma t|\mu x.t$$

Let $M = A \cup \{1'\}$. The following rules describe the possible transitions of closed terms:

1. $ap \xrightarrow{a} p$ for all $a \in A$ and all p
2. $p \xrightarrow{m} p' \Rightarrow p + q \xrightarrow{m} p' + q$
3. $t[\mu x.t/x] \xrightarrow{m} p \Rightarrow \mu x.t \xrightarrow{m} p$
4. **a** $\delta p \xrightarrow{1} \delta p$

   b $\varepsilon p \xrightarrow{1'} \epsilon p$

   c $\gamma p \xrightarrow{1} \gamma p$ if there does not exist $m \in M, p'$ such that $p \xrightarrow{m} p'$

   $\gamma p \xrightarrow{1'} \gamma p$ otherwise.

5. $p \xrightarrow{m} q$ implies

$$\delta p \xrightarrow{m^+} q$$

$$\varepsilon p \xrightarrow{m^+} q$$

$$\gamma p \xrightarrow{m^+} q$$

where $a^+ = a$ for $a \in A$ and $(1')^+ = 1$.

The following definition rules out as inadmissible, sequences with an infinite suffix of $1'$.

Definition 1. A sequence $p_0 \xrightarrow{m_1} p_1 \xrightarrow{m_2} \ldots$ is *inadmissible* if there exists some k such that $m_i = 1'$ for $i \geq k$. Otherwise a sequence is *admissible.*

The notion of admissible together with the transition rules for ε and γ determine whether infinite delay sequences are possible or not. It is worth noting the explicit context-sensitivity of γ.

Definition 2. For each closed term p, define $\mathcal{M}(p)$ to be the set of all admissible sequences from p, and all prefixes of such sequences. We take $p = q$ if $\mathcal{M}(p) = \mathcal{M}(q)$.

We would like to show that γ is not expressible in terms of δ and ε, i.e. there is no context $\mathcal{C}[\cdot]$ not involving γ such that $\mathcal{M}(\mathcal{C}[p]) = \mathcal{M}(\gamma p)$ for all closed terms p. We do this by showing that for any context $\mathcal{C}[\cdot]$, $\mathcal{M}(\mathcal{C}[\emptyset]) \subseteq \mathcal{M}(\mathcal{C}[p])$ for all closed p. Since the infinite sequence $1, 1, 1, \ldots$ is in $\mathcal{M}(\gamma\emptyset)$, but not in, say, $\mathcal{M}(\gamma a \emptyset)$ for $a \in A$, this effectively proves that γ is not representable. One can think of this as saying that contexts constructed out of ε and the other terms in the calculus satisfy a certain "monotonicity" property that is violated by γ.

Theorem 1. For any context $\mathcal{C}[\cdot]$, $\mathcal{M}(\mathcal{C}[\emptyset]) \subseteq \mathcal{M}(\mathcal{C}[p])$ for all closed terms p.

*Proof:*The proof proceeds by induction on the structure of the context.

1. Suppose $\mathcal{C}[\mathcal{X}] = t$ where t is $\mathcal{X}$-free. Then $\mathcal{M}(\mathcal{C}[p]) = \mathcal{M}(\mathcal{C}[t])$ for all closed p.

2. Suppose $\mathcal{C}[\mathcal{X}] = \mathcal{X}$. Then $\mathcal{M}(\mathcal{C}[\emptyset]) = \{\lambda\} \subseteq \mathcal{M}(\mathcal{C}[p])$ for all p, where λ stands for the empty string.

3. Suppose $\mathcal{C}[\mathcal{X}] = \varepsilon\mathcal{C}_1[\mathcal{X}]$ where the result holds for $\mathcal{C}_1[\cdot]$. Then for any closed p,

$$\begin{aligned}
\mathcal{M}(\varepsilon\mathcal{C}_1[\emptyset]) &= \{(1')^k a_1^+ a_2 \ldots a_n \text{ (or } (1')^k a_1^+ a_2 \ldots) \mid k \geq 0,\ n \geq 0 \text{ and} \\
&\qquad a_1 a_2 \ldots a_n \text{ (or } a_1 a_2 \ldots) \text{ is an element of } \mathcal{M}(\mathcal{C}_1[\emptyset])\} \\
&\subseteq \{(1')^k a_1^+ a_2 \ldots a_n \text{ (or } (1')^k a_1^+ a_2 \ldots) \mid k \geq 0,\ n \geq 0 \text{ and} \\
&\qquad a_1 a_2 \ldots a_n \text{ (or } a_1 a_2 \ldots) \text{ is an element of } \mathcal{M}(\mathcal{C}_1[p])\} \\
&= \mathcal{M}(\varepsilon\mathcal{C}_1[p]).
\end{aligned}$$

Thus $\mathcal{M}(\mathcal{C}[\emptyset]) \subseteq \mathcal{M}(\mathcal{C}[p])$ for any closed p.

4. Suppose $\mathcal{C}[\mathcal{X}] = \delta\mathcal{C}_1[\mathcal{X}]$, where the result holds for $\mathcal{C}_1[\mathcal{X}]$. The same argument as in case 3 above then establishes the result for $\mathcal{C}[\mathcal{X}]$.

5. Suppose $\mathcal{C}[\mathcal{X}] = a\mathcal{C}_1[\mathcal{X}]$, where $a \in A$ and the result holds for $\mathcal{C}_1[\mathcal{X}]$. Again, almost the same argument as in case 3 will prove the result for $\mathcal{C}[\mathcal{X}]$.

6. Suppose $\mathcal{C}[\mathcal{X}] = \mathcal{C}_1[\mathcal{X}] + \mathcal{C}_2[\mathcal{X}]$, where the result holds for $\mathcal{C}_1[\mathcal{X}]$ and $\mathcal{C}_2[\mathcal{X}]$. Then since $\mathcal{M}(\mathcal{C}_1[\mathcal{X}] + \mathcal{C}_2[\mathcal{X}]) = \mathcal{M}(\mathcal{C}_1[\mathcal{X}]) \cup \mathcal{M}(\mathcal{C}_2[\mathcal{X}])$, the result holds for $\mathcal{C}[\mathcal{X}]$ also.

7. Finally, suppose $\mathcal{C}[\mathcal{X}] = \mu x_0.t_0$ where t_0 contains at least one instance of $\mathcal{X}$.

   Claim 1. The sequences/prefixes from $\mu x_0.t_0[\emptyset/\mathcal{X}]$ will also be sequences/prefixes from $\mu x_0.t_0[p/\mathcal{X}]$ for all p, and will be produced by the same series of μ-expansions.

   Proof: We use induction on the number of μ-expansions used to produce a given admissible sequence or prefix.

   The term $t_0[\emptyset/\mathcal{X}]$ is a disjunction of terms of the following kinds:

   a. $s_1 s_2 \ldots s_k x_0$

b. $s_1 s_2 \ldots s_k(\mu x_{j_1}.t_{j_1})[\emptyset/\mathcal{X}]$

c. $t'_0[\emptyset/\mathcal{X}]$ where t'_0 is some μ-free term containing no variables other than $\mathcal{X}$.

d. $s_1 s_2 \ldots s_k(t'_0 + \ldots + t'_n)$ where each of $t'_0, \ldots, t'_n$ is, in turn, of one of the forms (a)-(d).

Here, k is some number ≥ 0, and s_i is either $\varepsilon, \delta,$ or some $a \in A$ for each i. Since for any closed terms t and t', $a(t+t') = at + at'$, and similarly for ε and δ, the subterms of $t_0[\emptyset/\mathcal{X}]$ of type (d) can be re-written as the sum of terms of type (a)-(c). Therefore, the sequences and prefixes of sequences from $\mu x_0.t_0[\emptyset/\mathcal{X}]$ are those from

a. $s_1 s_2 \ldots s_k(\mu x_0.t_0)[\emptyset/\mathcal{X}]$

b. $s_1 s_2 \ldots s_k(\mu x_{j_1}.t_{j_1})[\mu x_0.t_0/x_0][\emptyset/\mathcal{X}]$

c. $t'_0[\emptyset/\mathcal{X}]$.

In case (c), t'_0, being μ- and variable-free, must be a context falling under one of the six cases discussed above; therefore any sequence produced by $t'_0[\emptyset/\mathcal{X}]$ will also be produced by $t'_0[p/\mathcal{X}]$ for all closed p. Also, since we are only concerned with sequences resulting from one μ-expansion, the μ-term that follows the $s_1, \ldots, s_k$ in (a) and (b) has no bearing on the sequences produced, so that exactly the same ones will be produced with $\mathcal{X}$ replaced by any p. Thus, any sequence of actions produced with no μ-expansions from a disjunct of $\mu x_0.t_0[\emptyset/\mathcal{X}]$ will be a sequence from the corresponding disjunct of $\mu x_0.t_0[p/\mathcal{X}]$ for any p.

Suppose now that the result is true for sequences produced by n μ-expansions. Consider those produced by $n+1$ expansions. Again, the disjuncts of $t_0[\mu x_0.t_0/x_0][\emptyset/\mathcal{X}]$ are of the form

a. $s_1 s_2 \ldots s_k(\mu x_0.t_0)[\emptyset/\mathcal{X}]$

b. $s_1 s_2 \ldots s_k(\mu x_{j_1}.t_{j_1})[\mu x_0.t_0/x_0][\emptyset/\mathcal{X}]$

c. $t'_0[\emptyset/\mathcal{X}]$ where t'_0 is μ-free and contains no variables other than $\mathcal{X}$.

Now let S be any admissible sequence, or prefix thereof, produced from $\mu x_0.t_0[\emptyset/\mathcal{X}]$ by $n+1$ μ-expansions. Then S must have one of two forms:

(i) If S is produced by expanding $s_1s_2\ldots s_k(\mu x_{j_1}.t_{j_1})[\mu x_0.t_0/x_0][\emptyset/\mathcal{X}]$ where $s_k \in A$ or $k = 0$, then $S = S_1S_2$ where S_1 is produced by $s_1s_2\ldots s_k$ and ends with s_k; and S_2 is produced by the term $(\mu x_{j_1}.t_{j_1})[\mu x_0.t_0/x_0][\emptyset/\mathcal{X}]$.

(ii) If S is produced by expanding a term $(\mu x_{j_1}.t_{j_1})[\mu x_0.t_0/x_0][\emptyset/\mathcal{X}]$ where $s_k = \varepsilon$ or $s_k = \delta$, then $S = S_1m^+S_2$ where S_1 is produced by $s_1s_2\ldots s_k$ and mS_2 is produced by the subsequent μ-term.

In either case, the sequences produced by $s_1s_2\ldots s_k$ are independent of the term following, and by the induction hypothesis, any sequence produced by n unwindings from terms of type (a) or (b) can be duplicated by performing the same series of expansions from $\mu x_0.t_0[p/\mathcal{X}]$ or $(\mu x_{j_1}.t_{j_1})[\mu x_0.t_0/x_0][p/\mathcal{X}]$ for any p. Therefore, if S is a sequence from $s_1s_2\ldots s_k(\mu x_{j_1}.t_{j_1})[\mu x_0.t_0/x_0][\emptyset/\mathcal{X}]$ produced by n μ-expansions, then the same series of expansions from $s_1s_2\ldots s_k(\mu x_{j_1}.t_{j_1})[\mu x_0.t_0/x_0][p/\mathcal{X}]$ will produce the same sequence for any p. Thus the result holds for sequences produced from $\mu x_0.t_0$ in $n+1$ unwindings.

Finally, consider an admissible sequence S from $\mu x_0.t_0[\emptyset/\mathcal{X}]$ produced by an infinite number of μ-expansions. Suppose $\mu x_0.t_0[\emptyset/\mathcal{X}]$, $\sigma_1\mu x_{j_1}.t_{j_1}[\emptyset/\mathcal{X}]$, $\sigma_2\mu x_{j_2}.t_{j_2}[\emptyset/\mathcal{X}]\ldots$ are the terms which are expanded to produce the sequence, where each σ_i is some string of ε's, δ's and actions. It must be the case that each $\sigma_i\mu x_{j_i}.t_{j_i}[\emptyset/\mathcal{X}]$ is a disjunct of
$t_{j_{(i-1)}}[\mu x_{j_{(i-1)}}.t_{j_{(i-1)}}/x_{j_{(i-1)}}][\emptyset/\mathcal{X}]$. Therefore, $\mu x_0.t_0[p/\mathcal{X}]$, $\sigma_1\mu x_{j_1}.t_{j_1}[p/\mathcal{X}]$, $\sigma_2\mu x_{j_2}.t_{j_2}[p/\mathcal{X}],\ldots$ will be a series of terms such that $\sigma_i\mu x_{j_i}.t_{j_i}[p/\mathcal{X}]$ is a disjunct of $t_{j_{(i-1)}}[\mu x_{j_{(i-1)}}.t_{j_{(i-1)}}/x_{j_{(i-1)}}][p/\mathcal{X}]$. Since each term with p substituted in for $\mathcal{X}$ can produce any finite sequence produced by that term with $\emptyset$ substituted for $\mathcal{X}$, S will also be an admissible sequence from $\mu x_0.t_0[p/\mathcal{X}]$.

This completes the proof of the main theorem. ∎

We have shown, then, that in this simple system, it is impossible to reproduce the behaviour of γ by any γ-free context.

3 Delay Operators in SCCS

We switch our attention now to Milner's Synchronous Calculus of Communicating Systems itself. Here, $(Act, \times, ^{-}, 1)$ is an abelian group of actions. One defines *behaviour expressions* over this group by

$$E ::= X | a{:}E | \sum_{i \in I} E_i | E \times F | E\lceil A | fix_i\tilde{X}\tilde{E} | \varepsilon E$$

where $a \in Act$, A is a subset of Act containing 1, and I is a possibly infinite indexing set. The expression $fix_i\tilde{X}\tilde{E}$ stands for the i^{th} component of the solution to the system of equations $X_j = E_j$ for $j \in I$. The operators fix_i bind variables; the closed behaviour expressions are called *agents*.

The possible actions of behaviour expressions are given by the following rules:

1. $a{:}E \xrightarrow{a} E$ for all $a \in Act$ and all E
2. $E_i \xrightarrow{a} E' \Rightarrow \sum_{i \in I} E_i \xrightarrow{a} E'$
3. $E \xrightarrow{a} E'$ and $F \xrightarrow{b} F' \Rightarrow E \times F \xrightarrow{ab} E' \times F'$
4. $E \xrightarrow{a} E'$ and $a \in A \Rightarrow E\lceil A \xrightarrow{a} E'\lceil A$
5. $E_i[fix\tilde{X}\tilde{E}/\tilde{X}] \xrightarrow{a} E' \Rightarrow fix_i\tilde{X}\tilde{E} \xrightarrow{a} E'$
6. **a.** $\varepsilon E \xrightarrow{1} \varepsilon E$
   b. $E \xrightarrow{a} E' \Rightarrow \varepsilon E \xrightarrow{a} E'$

If u is the sequence $a_1, a_2, \ldots$ (or $a_1, \ldots, a_n$) where each a_i is in Act, then a *u-computation* of the agent P_0 is a sequence $P_0 \xrightarrow{a_1} P_1 \xrightarrow{a_2} \ldots$ (or $P_0 \xrightarrow{a_1} \ldots \xrightarrow{a_n} P_n$).

The delay operator ε is meant to model finite but unbounded delay (note that the delay operator δ that allows infinite delay in all circumstances can be defined using the *fix* construct: $\delta E \equiv fixX(1 : X + E)$ for all E). An *admissible u-computation* is one in which every agent of the form εP which progresses at some point by using rule (6a) must eventually progress using rule (6b).

One might wish to define a new delay operator γ, that could delay infinitely in some circumstances only. However, it is once again our assertion that it is impossible to build such a context-dependent delay operator from the other constructs in the language. The argument proceeds in very much the same way as the preceeding one.

For each agent P, we define $\mathcal{M}(P)$ to be the set of all sequences of actions u such that there is an admissible u-computation of P, and all prefixes of such sequences.

Claim 2. For any context $\mathcal{C}[\cdot]$, $\mathcal{M}(\mathcal{C}[\emptyset]) \subseteq \mathcal{M}(\mathcal{C}[P])$ for all agents P.

Proof: This is almost an exact repetition of the structural induction argument used in the proof of Claim 1, with only two additional cases.

1. If $\mathcal{C}[\mathcal{X}]$ is P (where P is $\mathcal{X}$-free), $\mathcal{X}$, $\varepsilon\mathcal{C}_1[\mathcal{X}]$ or $a{:}\mathcal{C}_1[\mathcal{X}]$ where $a \in Act$, then the argument is the same as that in the corresponding cases of Claim 1, the only difference being that in the ε case, $(1')^k$ is replaced now by 1^k.

2. If $\mathcal{C}[\mathcal{X}] = \sum_{i\in I} \mathcal{C}_i[\mathcal{X}]$, then $\mathcal{M}(\mathcal{C}[\mathcal{X}]) = \cup_{i\in I}\mathcal{M}(\mathcal{C}_i[\mathcal{X}])$, so if the result holds for each $\mathcal{C}_i[\cdot]$ then it holds for $\mathcal{C}[\cdot]$.

3. If $\mathcal{C}[\mathcal{X}] = \mathcal{C}_1[\mathcal{X}] \times \mathcal{C}_2[\mathcal{X}]$, and the claim is true of $\mathcal{C}_1[\cdot]$ and of $\mathcal{C}_2[\cdot]$, then

$$\begin{aligned}
\mathcal{M}(\mathcal{C}_1[\emptyset] \times \mathcal{C}_2[\emptyset]) &= \{c_1c_2\ldots c_n \text{ (or } c_1c_2\ldots)\mid c_i = a_ib_i \\
&\qquad \text{and } a_1a_2\ldots a_n \text{ (or } a_1a_2\ldots) \text{ is in } \mathcal{M}(\mathcal{C}_1[\emptyset]) \\
&\qquad \text{and } b_1b_2\ldots b_n \text{ (or } b_1b_2\ldots) \text{ is in } \mathcal{M}(\mathcal{C}_2[\emptyset])\} \\
&\subseteq \{c_1c_2\ldots c_n \text{ (or } c_1c_2\ldots)\mid c_i = a_ib_i \\
&\qquad \text{and } a_1a_2\ldots a_n \text{ (or } a_1a_2\ldots) \text{ is in } \mathcal{M}(\mathcal{C}_1[P]) \\
&\qquad \text{and } b_1b_2\ldots b_n \text{ (or } b_1b_2\ldots) \text{ is in } \mathcal{M}(\mathcal{C}_2[P])\} \\
&= \mathcal{M}(\mathcal{C}_1[P] \times \mathcal{C}_2[P])
\end{aligned}$$

4. If $\mathcal{C}[\mathcal{X}] = \mathcal{C}_1[\mathcal{X}]\lceil A$ where the result holds for $\mathcal{C}_1[\cdot]$, then

$$\begin{aligned}
\mathcal{M}(\mathcal{C}[\emptyset]) &= \{a_1a_2\ldots a_n \text{ (or } a_1a_2\ldots) \in \mathcal{C}_1[\emptyset] \mid \\
&\qquad a_i \in A \text{ for all } i\} \\
&\subseteq \{a_1a_2\ldots a_n \text{ (or } a_1a_2\ldots) \in \mathcal{C}_1[P] \mid \\
&\qquad a_i \in A \text{ for all } i\} \\
&= \mathcal{M}(\mathcal{C}[P])
\end{aligned}$$

5. Finally, if $\mathcal{C}[P] = fix_i\tilde{X}\tilde{E}$, then the proof of the result proceeds as in the final case of Claim 1, the induction being on the largest number of nested fix-expansions performed.

   Consider first the form of the expression E_i, the i^{th} component of $\tilde{E}$. Since a:, ε, and $\lceil A$ all distribute over $+$, E_i can be written as a disjunct of expressions of the general form

   a $s_1(s_2(\ldots(s_k(X_j\lceil A))\lceil A_k \ldots)\lceil A_2)\lceil A_1$ for some $j \in I, k \geq 0$

   b $s_1(s_2(\ldots(s_k((E' \times E'')\lceil A))\lceil A_k \ldots)\lceil A_2)\lceil A_1$ where E' and E'' may contain instances of $\mathcal{X}$ and $X_j, j \in I$

   c $P\lceil A$ where P is fix-free and contains no variables other than $\mathcal{X}$.

   (Here, s_j is either a: for some $a \in Act$, or ε.) Therefore any disjunct of $E_i[fix\tilde{X}\tilde{E}/\tilde{X}]$ has one of the following forms:

   a $s_1(s_2(\ldots(s_k(fix_j\tilde{X}\tilde{E})\lceil A))\lceil A_k \ldots)\lceil A_2)\lceil A_1$ for some $j \in I$

   b $s_1(s_2(\ldots(s_k((E' \times E'')\lceil A))\lceil A_k \ldots)\lceil A_2)\lceil A_1[fix\tilde{X}\tilde{E}/\tilde{X}]$

   c $P\lceil A$ where P is fix-free.

   Now consider the admissible sequences of actions from $E_i[fix\tilde{X}\tilde{E}/\tilde{X}][\emptyset/\mathcal{X}]$ that do not involve actions from $fix_j\tilde{X}\tilde{E}$ for any $j \in I$. These will be the same as those produced by replacing all instances of $fix_j\tilde{X}\tilde{E}$ by any agent that can perform no actions, for instance $\emptyset$. But replacing each $fix_j\tilde{X}\tilde{E}$ in the disjuncts of type (a) and (b) by $\emptyset$ produces contexts which fall into one of the four cases discussed above. Thus any sequence produced by substituting $\emptyset$ for $\mathcal{X}$ in such a context will be produced by substituting any agent P for $\mathcal{X}$. So the sequences of actions from $E_i[fix\tilde{X}\tilde{E}/\tilde{X}][\emptyset/\mathcal{X}]$ using no instances of rule (5) are also sequences from $E_i[fix\tilde{X}\tilde{E}/\tilde{X}][P/\mathcal{X}]$ for all agents P; i.e. the elements of $\mathcal{M}(\mathcal{C}[\emptyset])$ produced by invoking rule (5) once are also elements of $\mathcal{M}(\mathcal{C}[P])$ for all agents P.

   Suppose then that any admissible sequence of actions produced from $fix_i\tilde{X}\tilde{E}[\emptyset/\mathcal{X}]$ by unwinding series of no more than n nested fix-expressions is also produced by expanding the corresponding agents in $fix_i\tilde{X}\tilde{E}[P/\mathcal{X}]$ for all P. Let u be a sequence from $fix_i\tilde{X}\tilde{E}$ produced using rule (5) on $n+1$ nested expressions. Then u is produced by a disjunct of type (a) or (b) above using no more than n nested instances of rule (5), and so, by the inductive hypothesis, will also be a sequence from that disjunct with $\mathcal{X}$ replaced by any P. Thus u will be a sequence from $fix_i\tilde{X}\tilde{E}$.

Finally, in the case of an admissible sequence u produced using an infinite number of fix-expansions, following the computation from $fix_i\tilde{X}\tilde{E}[\emptyset/\mathcal{X}]$ and developing the corresponding expressions in $fix_i\tilde{X}\tilde{E}[P/\mathcal{X}]$ at each stage will produce an admissible u-computation. Therefore $u \in \mathcal{M}(\mathcal{C}[\emptyset])$ implies $u \in \mathcal{M}(\mathcal{C}[P])$ for any agent P.

4 Conclusions

In this note we have established the difference between two different finite delay operators that are considered in conjunction with SCCS. The interesting aspect of this proof is that it shows a situation with two different forms of unbounded indeterminacy. The proof technique that we use is conceptually similar to the proof of the corresponding result for dataflow given by Panangaden and Stark [4] for dataflow. Both proofs rely on a monotonicity property. The situation, however, is quite different in details. SCCS is a synchronous calculus. Even with the introduction of the delay operators the communication between processes is synchronous unlike the dataflow case. The proofs given in [4] use the fact that the network structure is static. In the present work we note that SCCS permits unwinding of recursively defined processes. Thus process structure may change to some extent. Finally, the equivalence that we use is trace equivalence, one of the weakest among those commonly considered. This is, of course, a good thing from the point of view of an inequivalence proof.

Acknowledgements

We acknowledge with pleasure helpful discussions with Joachim Parrow and Bard Bloom.

References

[1] M. Hennessy. Axiomatizing finite delay operators. *Acta. Inform.*, pages 61–88, 1984.

[2] R. Milner. A finite-delay operator in synchronous ccs. Technical Report CSR-116-82, University of Edinburgh, 1982.

[3] R. Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25:267–310, 1983.

[4] P. Panangaden and E. W. Stark. Computations, residuals and the power of indeterminacy. In Timo Lepisto and Arto Salomaa, editors, *Proceedings of the Fifteenth ICALP*, pages 439–454. Springer-Verlag, 1988. Lecture Notes in Computer Science 317.