

# Nondeterminism and divergence created by concealment in CSP

***Citation for published version (APA):***

Verhoeff, T. (1986). *Nondeterminism and divergence created by concealment in CSP*. (Computing science notes; Vol. 8606). Technische Universiteit Eindhoven.

***Document status and date:***

Published: 01/01/1986

***Document Version:***

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

***Please check the document version of this publication:***

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

***General rights***

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

***Take down policy***

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

Nondeterminism and Divergence  
Created by Concealment in CSP

by

Tom Verhoeff

86/06

December 1986

## COMPUTING SCIENCE NOTES

This is a series of notes of the Computing Science Section of the Department of Mathematics and Computing Science of Eindhoven University of Technology.

Since many of these notes are preliminary versions or may be published elsewhere, they have a limited distribution only and are not for review.

Copies of these notes are available from the author or the editor.

Eindhoven University of Technology  
Department of Mathematics and Computing Science  
P.O. Box 513  
5600 MB EINDHOVEN  
The Netherlands  
All rights reserved  
editor: F.A.J. van Neerven

# Nondeterminism and Divergence Created by Concealment in CSP

*Tom Verhoeff*

Department of Mathematics and Computing Science  
Eindhoven University of Technology  
Eindhoven, The Netherlands  
September 1986

## *ABSTRACT*

In [0] the notion of a *process* is defined. *Deterministic* processes form a special subclass of processes. Nondeterministic processes can be obtained from deterministic ones by combining them, for instance, with the choice operator. They can also be obtained by concealment. Apart from their kind of nondeterminism, processes are distinguished by their *divergences*. Divergent processes can be created from nondivergent ones by unguarded recursion or by concealment.

This paper shows that any CSP process can be obtained from a suitably chosen deterministic CSP process by concealment. The proof is given in the form of an explicit construction. We use the notation and some results of [1], especially relevant is Section 5.3. The reader is assumed to be familiar with CSP [0] and Trace Theory [1].

## **0 INTRODUCTION**

Trace Theory identifies processes that in CSP are distinguished. CSP uses a more refined notion of processes in order to express deadlock and livelock properties under a number of operators. A notoriously difficult operator in Trace Theory is projection, which corresponds to the concealment operator of CSP.

The question that triggered the investigation reported in this paper is: Which of the additional features of CSP processes can be considered as having been "caused" by projection? A related question is: In how far are the additional distinctions that CSP offers relevant from the point of view of Trace Theory?

The answer to the first question will be: All of them, since all varieties of CSP processes can be obtained by projecting suitable deterministic processes. The processes of Trace Theory can be identified with the deterministic CSP processes (see [1], Section 5.3). Hence, the second question is answered with: All the way, if one is interested in projection.

In Section 1 we have collected the most important definitions and a few general properties. Section 2 contains the major result of this paper and its proof in the form of a construction. This construction is applied to some examples in Section 3, while Section 4 presents some alternative constructions. In Section 5 the construction is explained further in terms of state graphs. Section 6 is the conclusion. The appendix presents a different—but for us more useful—formulation of the main theorem of Section 5.3 in [1].

## 1 PRELIMINARIES

This section establishes the background necessary to formulate the problem and our solution. It does not give additional explanatory interpretations of the definitions. The following notions are defined formally: TT-process (process in the sense of Trace Theory), CSP-process (possibly nondeterministic divergent process in the sense of CSP), correspondence between TT-processes and CSP-processes, successor set, deterministic CSP-process, projection of CSP-process (called concealment in CSP). Some auxiliary formulae for projection are derived.

A *TT-process* is a pair  $\langle A, V \rangle$  where

$A$  is a set of symbols (alphabet)

$V$  is a subset of  $A^*$  (traces)

such that

$\epsilon \in V$  (T0)

$tu \in V \Rightarrow t \in V$  (T1)

Let  $T$  be a TT-process. When  $T = \langle A, V \rangle$ , we write  $\mathbf{a}T$  for  $A$  and  $\mathbf{t}T$  for  $V$ . For trace  $t, t \in \mathbf{t}T$ , its *successor set*  $S(t, T)$  is defined by

$$S(t, T) = \{a \in \mathbf{a}T \mid ta \in \mathbf{t}T\}$$

A *CSP-process* is a triple  $\langle A, F, D \rangle$  where

$A$  is a set of symbols (alphabet)

$F$  is a set of pairs  $(t, X)$  where  $t \in A^*$  and  $X \subseteq A$  (failures)

$D$  is a subset of  $A^*$  (divergences)

such that

$(\epsilon, \emptyset) \in F$  (C0)

$(tu, X) \in F \Rightarrow (t, \emptyset) \in F$  (C1)

$(t, X) \in F \wedge Y \subseteq X \Rightarrow (t, Y) \in F$  (C2)

$(t, X) \in F \wedge a \in A \Rightarrow (t, X \cup \{a\}) \in F \vee (ta, \emptyset) \in F$  (C3)

$$D \subseteq \{t \mid (t, \emptyset) \in F\} \quad (C4)$$

$$t \in D \wedge u \in A^* \Rightarrow tu \in D \quad (C5)$$

$$t \in D \wedge X \subseteq A \Rightarrow (t, X) \in F \quad (C6)$$

Notice that C4 is actually superfluous since C6 implies C4. For a pair  $(t, X)$ ,  $(t, X) \in F$ , we call  $X$  a *refusal set* of  $t$ .

Let  $P$  be a CSP-process. When  $P = \langle A, F, D \rangle$ , we write  $\mathbf{a}P$  for  $A$ ,  $\mathbf{f}P$  for  $F$ , and  $\mathbf{d}P$  for  $D$ .

For a CSP-process  $P$  the corresponding TT-process is denoted by  $tr(P)$  and is defined by

$$tr(P) = \langle \mathbf{a}P, \{t \mid (t, \emptyset) \in \mathbf{f}P\} \rangle$$

We write  $\mathbf{t}P$  as short for  $\mathbf{t}tr(P)$ ; C4, for instance, expresses that  $\mathbf{d}P \subseteq \mathbf{t}P$ . The successor set  $S(t, tr(P))$  is also denoted by  $S(t, P)$ . From the definition of  $tr$  we see that

$$S(t, P) = \{a \in \mathbf{a}P \mid (ta, \emptyset) \in \mathbf{f}P\}$$

A consequence of C3, which we shall need later on, is

$$(t, X) \in F \Rightarrow (t, X \cup (A \setminus S(t, P))) \in F \quad (C3a)$$

Let  $T$  be a TT-process. Following [1] the CSP-process corresponding to  $T$  is denoted by  $pr(T)$  and is defined by

$$pr(T) = \langle \mathbf{a}T, \{(t, X) \mid t \in \mathbf{t}T \wedge X \subseteq \mathbf{a}T \setminus S(t, T)\}, \emptyset \rangle$$

The conjunct  $X \subseteq \mathbf{a}T \setminus S(t, T)$  is equivalent to  $X \subseteq \mathbf{a}T \wedge X \cap S(t, T) = \emptyset$  and also to  $X \subseteq \mathbf{a}T \wedge S(t, T) \subseteq \mathbf{a}T \setminus X$ . Therefore we have

$$pr(T) = \langle \mathbf{a}T, \{(t, X) \mid t \in \mathbf{t}T \wedge X \subseteq \mathbf{a}T \wedge S(t, T) \subseteq \mathbf{a}T \setminus X\}, \emptyset \rangle \quad (PR)$$

From the definitions of  $pr$  and  $tr$  it follows that  $tr$  is a left inverse of  $pr$ , i.e.

$$tr(pr(T)) = T \quad (INV)$$

A CSP-process  $P$  is called *divergent* when  $\mathbf{d}P \neq \emptyset$ .  $P$  is called *deterministic* when

$$(\mathbf{A} t, X : : (t, X) \in \mathbf{f}P \equiv X \subseteq \mathbf{a}P \setminus S(t, P)) \wedge \mathbf{d}P = \emptyset$$

that is, when (i) it may refuse only those actions that it cannot engage in anyway and (ii) it is not divergent. In the light of C5 and C6 the second conjunct may be dropped when  $\mathbf{a}P \neq \emptyset$ , because in that case it is implied by the first conjunct. Determinism of  $P$  can also be expressed more concisely by

$$P = pr(tr(P)) \quad (DET)$$

For every TT-process  $T$ ,  $pr(T)$  is deterministic on account of INV. We also see that every deterministic CSP-process  $P$  corresponds to some TT-process, viz.  $tr(P)$ . The class of deterministic CSP-processes can, therefore, be identified with the class of TT-processes.

Let  $P$  be a CSP-process and let  $B$  be a subset of  $\mathbf{a}P$ . Define  $C$  as the complement of  $B$  with respect to  $\mathbf{a}P$ , that is  $C = \mathbf{a}P \setminus B$ . The *projection* of  $P$  on  $B$ , denoted by  $P \upharpoonright B$ , is the CSP-process  $\langle A_B, F_B, D_B \rangle$  where

$$\begin{aligned} A_B &= B \\ F_B &= \{(u, X) \mid u \in D_B \wedge X \subseteq B\} \cup \\ &\quad \{(t \upharpoonright B, X) \mid (t, X \cup C) \in \mathbf{f}P \wedge X \subseteq B\} \\ D_B &= (\mathbf{d}P \upharpoonright B) \cup \\ &\quad \{(t \upharpoonright B) \nu \mid (\mathbf{A} n : n \geq 0 : (\mathbf{E} s : s \in C^* \wedge l(s) > n : ts \in \mathbf{t}P)) \wedge \nu \in B^*\} \end{aligned}$$

In the terminology of [0] this is called  $P$  *without*  $C$ , written as  $P \setminus C$ . It corresponds to the *concealment* or *hiding* of events of  $C$  in the process  $P$ .

Let  $T$  be a TT-process and let  $B$  be a subset of  $\mathbf{a}T$ . We now compute  $pr(T) \upharpoonright B$ . Write  $pr(T) \upharpoonright B$  as  $\langle A_B, F_B, D_B \rangle$ . Define  $C$  as  $\mathbf{a}T \setminus B$ . Notice that for  $X \subseteq B$ ,  $\mathbf{a}T \setminus (X \cup C)$  equals  $B \setminus X$ . Substituting PR in the equations for projection yields

$$A_B = B \tag{H0}$$

$$\begin{aligned} F_B &= \{(u, X) \mid u \in D_B \wedge X \subseteq B\} \cup \\ &\quad \{(t \upharpoonright B, X) \mid t \in \mathbf{t}T \wedge X \subseteq B \wedge \mathbf{S}(t, T) \subseteq B \setminus X\} \end{aligned} \tag{H1}$$

$$D_B = \{(t \upharpoonright B) \nu \mid (\mathbf{A} n : n \geq 0 : (\mathbf{E} s : s \in C^* \wedge l(s) > n : ts \in \mathbf{t}T)) \wedge \nu \in B^*\} \tag{H2}$$

### Example

The alphabets  $A$  and  $B$ , and the TT-process  $T$  are defined by

$$\begin{aligned} A &= \{a, b, c\} \\ B &= \{a, b\} \\ T &= \langle A, \{\epsilon, a, c, cb\} \rangle \end{aligned}$$

The CSP-process corresponding to  $T$ , and the projection of this CSP-process on  $B$ , denoted by  $P_0$ , are given by

$$\begin{aligned} pr(T) &= \langle A, \text{clos}\{(\epsilon, \{b\}), (a, A), (c, \{a, c\}), (cb, A)\}, \emptyset \rangle \\ P_0 &= pr(T) \upharpoonright B = \langle B, \text{clos}\{(\epsilon, \{a\}), (a, B), (b, B)\}, \emptyset \rangle \end{aligned}$$

where *clos* is the closure operator on failure sets defined by

$$\text{clos}(F) = \{(t, Y) \mid (\mathbf{E} X : Y \subseteq X : (t, X) \in F)\} \tag{CLOS}$$

The TT-process  $T \upharpoonright B$  and its corresponding CSP-process are given by

$$T \upharpoonright B = \langle B, \{\epsilon, a, b\} \rangle$$

$$pr(T \upharpoonright B) = \langle B, clos\{(\epsilon, \emptyset), (a, B), (b, B)\}, \emptyset \rangle$$

Notice that here we have  $tr(P_0) = T \upharpoonright B$ , but that  $P_0 \neq pr(T \upharpoonright B)$ . Consequently,  $P_0$  is not deterministic.

In the notation for CSP used in [0] the above three CSP-processes would be written as follows.

$$pr(T) = (a \rightarrow STOP) | (c \rightarrow (b \rightarrow STOP))$$

$$P_0 = ((a \rightarrow STOP) | (b \rightarrow STOP)) \square (b \rightarrow STOP)$$

$$pr(T \upharpoonright B) = (a \rightarrow STOP) | (b \rightarrow STOP)$$

(End of Example)

## 2 CONSTRUCTION

Given is a CSP-process  $Q$ ,  $Q = \langle B, G, E \rangle$ . The problem is to find—if possible—a deterministic CSP-process  $P$  for which  $P \upharpoonright B = Q$ . Since each deterministic CSP-process corresponds to a TT-process, we construct a TT-process  $T$  with  $B \subseteq aT$  such that

$$pr(T) \upharpoonright B = Q \tag{CLAIM}$$

The construction of  $T$  is carried out in two stages. First it is assumed that  $E = \emptyset$ ; that is,  $Q$  is not divergent. In the second stage the construction is extended to divergent  $Q$ .  $T$  is not uniquely defined by CLAIM. Our construction is only one way of defining a suitable  $T$ . In Section 4 we shall discuss some alternatives.

### 2.0 First Stage

Assume  $E = \emptyset$ . Let  $c_X$ , for  $X \subseteq B$ , be a fresh symbol and define alphabet  $C$  as  $\{c_X \mid X \subseteq B\}$ ; we then have  $B \cap C = \emptyset$ . The intuitive meaning of symbol  $c_X$  will be that immediately after its occurrence the process refuses symbols from  $X$ . When  $C$  is hidden it means that the process *may* refuse symbols from  $X$  in such a state.

For  $aT$  we take  $B \cup C$ , and  $tT$  is defined inductively by

$$\epsilon \in tT \tag{P0}$$

$$\text{For } t \in tT \text{ where } t \text{ does not end in a symbol of } C, \text{ and } (t \upharpoonright B, X) \in G : tc_X \in tT \tag{P1}$$

$$\text{For } t \in tT \text{ where } t \text{ ends in symbol } c_X, \text{ and } b \in S(t \upharpoonright B, Q) \setminus X : tb \in tT \tag{P2}$$

That this is a sound definition requires a proof. We shall not cover all details, but we do make the following remarks in support of the definition.

The cases P0, P1, and P2 are mutually exclusive; that is, a trace is a member of  $tT$  on account of exactly one of P0, P1, and P2. For all traces  $t$ ,  $t \in tT$ , we have  $t \upharpoonright B \in tQ$ ; so



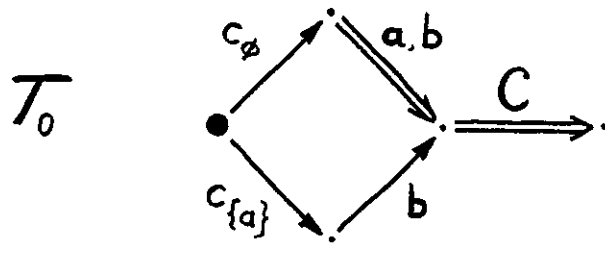
$S(t \upharpoonright B, Q)$  in P2 is well-defined. Notice that  $T$  is indeed a TT-process: T0 is implied by P0, and T1 is satisfied because P1 and P2 extend traces with a single symbol only.

Because  $S(t \upharpoonright B, Q) \subseteq B$  we see that symbols of  $B$  and  $C$  alternate in the traces of  $tT$ , not starting with a symbol of  $B$ . This is concisely expressed by

$$T \subseteq \text{SEM}_1(C, B) \tag{ALT}$$

**Example**

Below is given the state graph of the TT-process  $T_0$  that the construction would yield when applied to the CSP-process  $P_0$  of the example in Section 1. A double arrow labeled with a list or a set of symbols is an abbreviation for "parallel" arrows, each labeled with a symbol from that list or set.



(End of Example)

Let us return to the correctness proof for the construction. Again writing  $pr(T) \upharpoonright B$  as  $\langle A_B, F_B, D_B \rangle$  our proof obligation for CLAIM reduces to

$$A_B = B$$

$$F_B = G$$

$$D_B = \emptyset$$

The first of these is trivially fulfilled (see H0). The third line follows from H2 and ALT. Before proving the second line we derive three properties of the successor sets of  $T$ .

Let  $t$  be a trace in  $tT$  that does not end in a symbol of  $C$ . Distinguish two cases. If  $t = \epsilon$  then  $(t \upharpoonright B, \emptyset) = (\epsilon, \emptyset) \in G$  by C0. Otherwise suppose  $t = t_0 b$  with  $b \in B$ . Observe that  $t_0 b \in tT$  can only hold on account of P2. Therefore  $b \in S(t_0 \upharpoonright B, Q)$  holds, that is  $((t_0 \upharpoonright B) b, \emptyset)$  is a member of  $G$ . Since  $(t_0 \upharpoonright B) b = t_0 b \upharpoonright B = t \upharpoonright B$ , we have  $(t \upharpoonright B, \emptyset) \in G$ . P1 can, therefore, be applied to  $t$  with  $X = \emptyset$ . This shows that

For  $t \in tT$  where  $t$  does not end in a symbol of  $C$

$$c_0 \in S(t, T) \tag{S0}$$

On account of P1 we have

For  $t \in \mathfrak{t}T$  where  $t$  does not end in a symbol of  $C$

$$S(t, T) = \{c_X \mid (t \uparrow B, X) \in G\} \quad (S1)$$

And similarly, on account of P2, we have

For  $t \in \mathfrak{t}T$  where  $T$  ends in symbol  $c_X$  for some  $X \subseteq B$

$$S(t, T) = S(t \uparrow B, Q) \setminus X \subseteq B \setminus X \quad (S2)$$

Now we are ready to prove  $F_B = G$ . First we show that  $F_B \subseteq G$ .

Assume  $(u, X) \in F_B$  with  $X \subseteq B$ . Since  $D_B = \emptyset$  there is (see H1) a trace  $t, t \in \mathfrak{t}T$ , such that  $u = t \uparrow B$  and  $S(t, T) \subseteq B \setminus X$ . From S0 and  $S(t, T) \subseteq B$  we infer that  $t$  ends in a symbol of  $C$ . Say  $t = t_0 c_Y$  for some  $Y \subseteq B$ . We now derive

$$\begin{aligned} & S(t, T) \subseteq B \setminus X \\ \Rightarrow & \quad \{ \text{set calculus} \} \\ & X \cap S(t, T) = \emptyset \\ \equiv & \quad \{ S2, \text{ using } t = t_0 c_Y \} \\ & X \cap (S(t \uparrow B, Q) \setminus Y) = \emptyset \\ \equiv & \quad \{ \text{set calculus, using } X \subseteq B; t \uparrow B = u \} \\ & X \subseteq Y \cup (B \setminus S(u, Q)) \end{aligned}$$

Finally we derive

$$\begin{aligned} & t_0 c_Y \in \mathfrak{t}T \\ \Rightarrow & \quad \{ P1 \} \\ & (t_0 \uparrow B, Y) \in G \\ \equiv & \quad \{ \text{property of projection for traces} \} \\ & (t_0 c_Y \uparrow B, Y) \in G \\ \equiv & \quad \{ t = t_0 c_Y \} \\ & (t \uparrow B, Y) \in G \\ \equiv & \quad \{ u = t \uparrow B \} \\ & (u, Y) \in G \\ \Rightarrow & \quad \{ C3a \} \\ & (u, Y \cup (B \setminus S(u, Q))) \in G \\ \Rightarrow & \quad \{ C2, \text{ using } X \subseteq Y \cup (B \setminus S(u, Q)) \text{ as derived above} \} \\ & (u, X) \in G \end{aligned}$$

This concludes the proof of  $F_B \subseteq G$ . We proceed by showing by mathematical induction on the length of trace  $u$  that  $(u, X) \in G$  implies  $(u, X) \in F_B$ .

Let  $(u, X) \in G$  with  $X \subseteq B$ . On account of H1 and  $D_B = \emptyset$  our proof obligation, i.e.  $(u, X) \in F_B$ , reduces to exhibiting a trace  $t, t \in \mathfrak{t}T$ , such that  $t \uparrow B = u$  and

$S(t, T) \subseteq B \setminus X$ .

**Base**  $u = \epsilon$ . Take  $t = c_X$ . We derive

$$\begin{aligned}
 & \text{true} \\
 \equiv & \quad \{ P0 \} \\
 & \epsilon \in \mathbf{t}T \\
 \Rightarrow & \quad \{ P1, \text{ using } (\epsilon, X) = (u, X) \in G \} \\
 & c_X \in \mathbf{t}T \\
 \equiv & \quad \{ t = c_X \} \\
 & t \in \mathbf{t}T
 \end{aligned}$$

From S2 we infer  $S(t, T) \subseteq B \setminus X$ , hence, by H1, we have  $(t \uparrow B, X) \in F_B$ . Since  $u = \epsilon = c_X \uparrow B = t \uparrow B$ , we have  $(u, X) \in F_B$ .

**Step**  $u = u_0b$  with  $b \in B$ . We derive

$$\begin{aligned}
 & (u_0b, X) \in G \\
 \Rightarrow & \quad \{ C1 \} \\
 & (u_0, \emptyset) \in G \\
 \Rightarrow & \quad \{ \text{induction hypothesis} \} \\
 & (u_0, \emptyset) \in F_B
 \end{aligned}$$

Also taking into account H1 and  $D_B = \emptyset$ , there exists a trace  $t_0 \in \mathbf{t}T$  with  $u_0 = t_0 \uparrow B$  and  $S(t_0, T) \subseteq B$ . From the latter together with S0 it follows that  $t_0$  ends in  $c_Y$  for some  $Y \subseteq B$ , say  $t_0 = t_1c_Y$ . Define  $t = t_1c_0bc_X$ . Notice that  $b \in S(u_0, Q)$ , since  $(u_0b, \emptyset) = (u, \emptyset) \in G$  by C1. Now we derive

$$\begin{aligned}
 & t_0 \in \mathbf{t}T \\
 \equiv & \quad \{ t_0 = t_1c_Y \} \\
 & t_1c_Y \in \mathbf{t}T \\
 \Rightarrow & \quad \{ T1 \} \\
 & t_1 \in \mathbf{t}T \\
 \Rightarrow & \quad \{ S0, \text{ using that } t_1 \text{ does not end in a symbol of } C \} \\
 & t_1c_0 \in \mathbf{t}T \\
 \Rightarrow & \quad \{ P2, \text{ using } b \in S(u_0, Q) = S(t_1c_0 \uparrow B, Q) \setminus \emptyset \} \\
 & t_1c_0b \in \mathbf{t}T \\
 \Rightarrow & \quad \{ P1, \text{ using } (t_1c_0b \uparrow B, X) = (u_0b, X) = (u, X) \in G \} \\
 & t_1c_0bc_X \in \mathbf{t}T \\
 \equiv & \quad \{ \text{choice of } t \} \\
 & t \in \mathbf{t}T
 \end{aligned}$$

By S2 we have  $S(t, T) \subseteq B \setminus X$ . Combining this with  $t \in \mathbf{t}T$  and using H1 yields  $(u, X) = (t \uparrow B, X) \in F_B$ .

We have now shown that  $G \subseteq F_B$ , which completes the proof of  $F_B = G$ .

## 2.1 Second Stage

For a nondivergent CSP-process we have presented a construction of a deterministic CSP-process from which it can be obtained by concealment. We now consider the general case. Let  $Q, Q = \langle B, G, E \rangle$ , be a CSP-process and assume that  $S$  is a TT-process such that  $pr(S) \upharpoonright B = \langle B, G, \emptyset \rangle$ . The TT-process  $T$  is defined by

$$T = \langle aS \cup \{c\}, tS \cup \{tc^n \mid t \in tS \wedge t \upharpoonright B \in E \wedge n > 0\} \rangle$$

where  $c$  is a fresh symbol,  $c \notin aS$ . Notice that  $T \upharpoonright aS = S$ , and that for trace  $t, t \in tT$ ,

$$S(t, T) = \begin{cases} S(t, S) & \text{if } t \in tS \text{ and } t \upharpoonright B \notin E \\ S(t, S) \cup \{c\} & \text{if } t \in tS \text{ and } t \upharpoonright B \in E \\ \{c\} & \text{if } t \notin tS \end{cases} \quad (S3)$$

We prove that  $pr(T) \upharpoonright B = Q$ . Write  $pr(T) \upharpoonright B = \langle A_B, F_B, D_B \rangle$ . Our proof obligation reduces to

$$\begin{aligned} A_B &= B \\ F_B &= G \\ D_B &= E \end{aligned}$$

The alphabets are correct on account of H0. We prove  $D_B = E$ .

$(E \subseteq D_B)$  Let  $u \in E$ . From C4 we infer  $(u, \emptyset) \in G$ . Since  $G$  is also the failure set of the nondivergent process  $pr(S) \upharpoonright B$ , there exists (see H1) a trace  $t, t \in tS$ , such that  $u = t \upharpoonright B$ . For any  $n, n > 0$ , we have  $tc^n \in tT$  and, hence, by H2,  $u = t \upharpoonright B \in D_B$ .

$(D_B \subseteq E)$  Let  $u \in D_B$ . On account of H2 we have  $u = (t \upharpoonright B) \nu$  where  $\nu \in B^*$  and

$$(A \ n : n \geq 0 : (E \ s : s \in (aS \setminus B)^* \wedge l(s) > n : ts \in tT)) \quad (2.1.0)$$

Because  $pr(S) \upharpoonright B$  is not divergent  $u \notin d(pr(S) \upharpoonright B)$ . Hence (by H2) there is a number  $N, N \geq 0$ , such that

$$(A \ s : s \in (aS \setminus B)^* \wedge (t \upharpoonright aS) s \in tS : l(s) \leq N) \quad (2.1.1)$$

Taking  $n = N$ , let  $s$  be a trace according to (2.1.0). Since  $T \upharpoonright aS = S$  and using (2.1.1) we see that  $l(s \upharpoonright aS) \leq N < l(s)$ . Therefore  $s$  contains a symbol from  $(aS \setminus B) \setminus aS$ , that is,  $s$  contains the symbol  $c$ . This proves the existence of a trace  $s_0, s_0 \in (aS \setminus B)^*$ , such that  $ts_0c \in tT$ . From the construction of  $T$  and  $c \notin aS$  we then infer  $t \upharpoonright B = ts_0c \upharpoonright B \in E$ . Hence, by C5, we have  $u = (t \upharpoonright B) \nu \in E$ .

Finally we prove  $F_B = G$ .

$(F_B \subseteq G)$  Let  $(u, X) \in F_B$ . Distinguish two cases.

**Case  $u \in D_B$ .** From  $D_B = E$  (see above) and C6 we infer  $(u, X) \in G$ .

**Case  $u \notin D_B$ .** By H1 we can take  $t \in \mathfrak{t}T$  such that  $u = t \upharpoonright B$  and  $S(t, T) \subseteq B \setminus X$ . Since  $t \upharpoonright B = u \notin D_B = E$ , we have  $t \in \mathfrak{t}S$ . Looking at S3 we see that  $S(t, S) = S(t, T)$ . Hence (by H1)  $(t \upharpoonright B, X) \in G$ ,  $G$  being also the failure set of  $pr(S) \upharpoonright B$ .

( $G \subseteq F_B$ ) Let  $(u, X) \in G$ . Since  $pr(S) \upharpoonright B = \langle B, G, \emptyset \rangle$  we have, by H1,  $u = t \upharpoonright B$  for some  $t \in \mathfrak{t}S$  with  $S(t, S) \subseteq B \setminus X$ .

**Case  $u \in E = D_B$ .** Then  $(u, X) \in F_B$  by H1.

**Case  $u \notin E$ .** Then  $S(t, T) = S(t, S) \subseteq B \setminus X$ . From H1 we now infer  $(u, X) \in F_B$ .

This concludes the constructive proof of the

### Projection Theorem

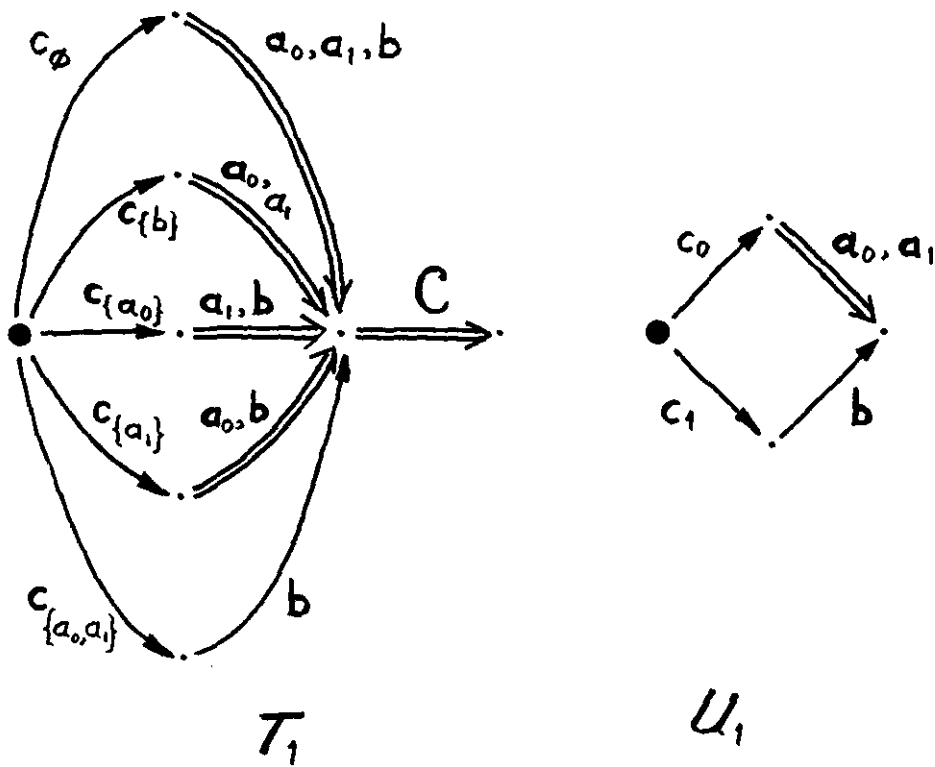
For every CSP-process  $Q$  there exists a deterministic CSP-process  $P$  such that  $P \upharpoonright \mathbf{a}Q = Q$ .

### 3 EXAMPLES

Before presenting variations of our construction we give some examples. Consider the CSP-process  $P_1$  defined by

$$P_1 = \langle \{a_0, a_1, b\} \\ , \text{clos} \{ (\epsilon, \{a_0, a_1\}), (\epsilon, \{b\}), (a_0, \mathbf{a}P_1), (a_1, \mathbf{a}P_1), (b, \mathbf{a}P_1) \} \\ , \emptyset \\ \rangle$$

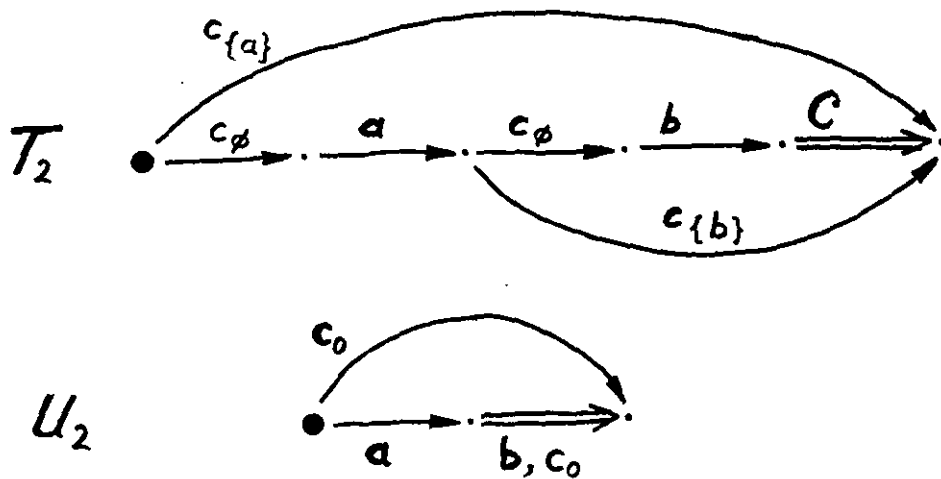
The state graphs for TT-processes  $T_1$  and  $U_1$  are given below.  $T_1$  is the result of the construction of Section 2 applied to  $P_1$ , and  $U_1$  is a different TT-process with  $pr(U_1) \upharpoonright \mathbf{a}P_1 = P_1$ .



Next consider the CSP-process  $P_2$  given by

$$P_2 = \langle \{a, b\}, \text{clos} \{(\epsilon, aP_2), (a, aP_2), (b, aP_2)\}, \emptyset \rangle$$

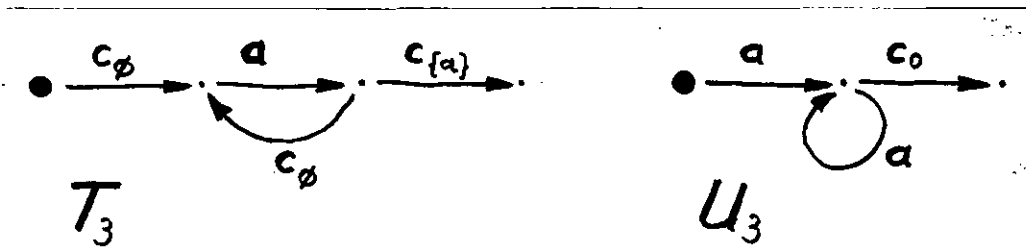
Applying the construction to  $P_2$  yields TT-process  $T_2$ ; an alternative with smaller alphabet is TT-process  $U_2$ .



Before giving an example with a divergent CSP-process we present the "almost divergent" CSP-process  $P_3$ :

$$\begin{aligned}
 P_3 = & \langle \{a\} \\
 & . \{(a^n, X) \mid n \geq 0 \wedge X \subseteq \{a\} \wedge (n = 0 \Rightarrow X = \emptyset)\} \\
 & . \emptyset \\
 & \rangle
 \end{aligned}$$

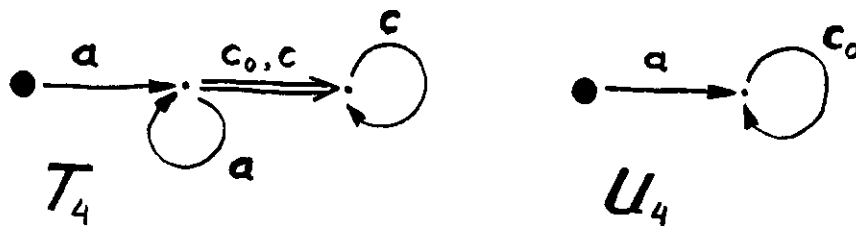
Again we give two TT-processes:  $T_3$  as obtained by the construction and a slightly simplified  $U_3$  for which also  $pr(U_3) \upharpoonright aP_3 = P_3$ .



Finally, we come to the divergent CSP-process  $P_4$ :

$$\begin{aligned}
 P_4 = & \langle \{a\} \\
 & . \{(a^n, X) \mid n \geq 0 \wedge X \subseteq \{a\} \wedge (n = 0 \Rightarrow X = \emptyset)\} \\
 & . \{a^n \mid n \geq 1\} \\
 & \rangle
 \end{aligned}$$

Notice that  $aP_3 = aP_4$  and  $fP_3 = fP_4$ . TT-process  $T_4$  was obtained by applying the second stage of the construction to TT-process  $U_3$  of the previous example. TT-process  $U_4$  is also satisfactory.



#### 4 ALTERNATIVE CONSTRUCTIONS

In this section we point out some variations on our construction that, in general, give rise to TT-processes with smaller alphabets and trace sets. The proofs are slightly more complicated and have been omitted.

Let us start with the first stage. Consider the extra symbol  $c_0$ . The alphabet  $\mathbf{a}T \setminus \{c_0\}$ , which we shall denote by  $A$ , is transparent (see Appendix) with respect to  $T$  on account of S0 and S2. Therefore,  $T \upharpoonright A$  also satisfies our requirement, since

$$\begin{aligned}
 & pr(T \upharpoonright A) \upharpoonright B \\
 = & \quad \{ \text{Transparency Theorem (see Appendix), using transparency of } A \} \\
 & (pr(T) \upharpoonright A) \upharpoonright B \\
 = & \quad \{ \text{property of projection, using } B \subseteq A \} \\
 & pr(T) \upharpoonright B \\
 = & \quad \{ \text{construction of } T \} \\
 & Q
 \end{aligned}$$

For the TT-process  $T \upharpoonright A$  property ALT does not hold, since S1 is no longer valid, but S2 remains true. The trace set of  $T \upharpoonright A$  can also be defined inductively by replacing P1 with P1a:

$$\begin{aligned}
 & \text{For } t \in \mathbf{t}T \text{ where } T \text{ does not end in a symbol of } C : \\
 & \text{if } (t \upharpoonright B, X) \in \mathbf{f}Q \wedge X \neq \emptyset \text{ then } tc_X \in \mathbf{t}T, \text{ and} \quad (P1a) \\
 & \text{if } b \in S(t \upharpoonright B, Q) \text{ then } tb \in \mathbf{t}T
 \end{aligned}$$

A further reduction of the trace set is possible by considering for each trace only its maximal nonempty refusal sets that are a subset of the successor set. For trace  $u$ ,  $u \in \mathbf{t}Q$ , these refusal sets are collected in the set  $M(u, Q)$  as expressed by the following definition.

$$\begin{aligned}
 M(u, Q) = & \{ X \mid (u, X) \in \mathbf{f}Q \wedge \emptyset \neq X \subseteq S(u, Q) \wedge \\
 & (\mathbf{A} Y : (u, Y) \in \mathbf{f}Q \wedge X \subseteq Y \subseteq S(u, Q) : X = Y) \}
 \end{aligned}$$

For trace  $u$ ,  $u \in \mathbf{t}Q$ , we then have

$$\begin{aligned}
 & \{ X \mid (u, X) \in \mathbf{f}Q \} \\
 = & \quad \{ C2 \text{ and } C3a, \text{ using } u \in \mathbf{t}Q \} \\
 & \{ X \mid X = \emptyset \vee (\mathbf{E} Y : Y \in M(u, Q) : X \subseteq Y \cup (\mathbf{a}Q \setminus S(u, Q))) \}
 \end{aligned}$$

That is,  $u$ 's refusal sets are completely determined by  $S(u, Q)$  and  $M(u, Q)$ . The intended additional reduction is accomplished by replacing P1 with P1b:

$$\begin{aligned}
 & \text{For } t \in \mathbf{t}T \text{ where } T \text{ does not end in a symbol of } C : \\
 & \text{if } X \in M(t \upharpoonright B, Q) \text{ then } tc_X \in \mathbf{t}T, \text{ and} \quad (P1b) \\
 & \text{if } b \in S(t \upharpoonright B, Q) \text{ then } tb \in \mathbf{t}T
 \end{aligned}$$



Notice that when using the construction based on P1b a trace is not extended with a symbol of  $C$  when it cannot be extended with symbols of  $B$ , since  $S(u, Q) = \emptyset$  implies  $M(u, Q) = \emptyset$ .

The last reduction for the first stage that we present is based on the following observation. A trace  $t$ ,  $t \in \mathbf{t}T$ , that does not end in a symbol of  $C$  need only be extended with those symbols of  $S(t \upharpoonright B, Q)$  that do not occur as successors after the extension of  $t$  with some symbol of  $C$ . Recall that the successors of  $tc_X$  are given by S2 as  $S(tc_X \upharpoonright B, Q) \setminus X$ , which equals  $S(t \upharpoonright B, Q)$  since  $c_X \notin B$ . Now define for trace  $u$ ,  $u \in \mathbf{t}Q$ , the set  $R(u, Q)$  as

$$R(u, Q) = S(u, Q) \setminus (\cup X : X \in M(u, Q) : S(u, Q) \setminus X)$$

We can simplify this definition by applying De Morgan's Law:

$$R(u, Q) = (\cap X : X \in M(u, Q) : X)$$

where an intersection over an empty range is taken to be  $S(u, Q)$ . All reductions are thrown together in P1c:

For  $t \in \mathbf{t}T$  where  $t$  does not end in a symbol of  $C$ :

if  $X \in M(t \upharpoonright B, Q)$  then  $tc_X \in \mathbf{t}T$ , and (P1c)

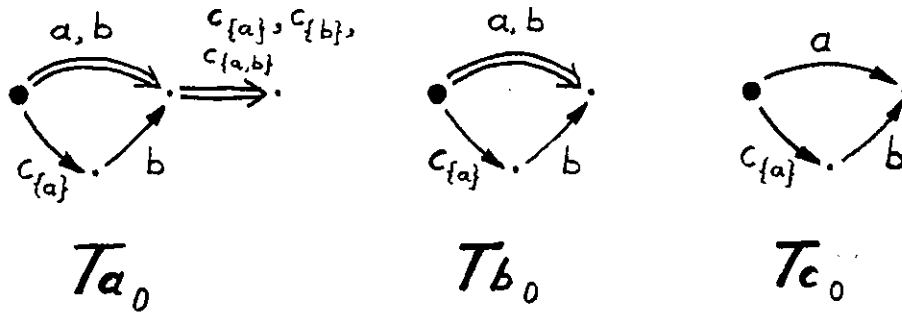
if  $b \in R(t \upharpoonright B, Q)$  then  $tb \in \mathbf{t}T$

With any of the modified definitions proposed above the symbol  $c_0$  does not occur in the traces of  $\mathbf{t}T$ . Symbols that never occur can be removed from  $\mathbf{a}T$  without affecting the projection of the CSP-process corresponding to  $T$ . In fact, the alphabet  $C$  can often be reduced even further. Its size need not exceed

$$(MAX t : t \in \mathbf{t}T : \#(S(t, T) \setminus B))$$

since the only function of the symbols in  $C$  is to distinguish among the possible extensions of each trace separately, they do not relate the extensions of different traces. For an example see CSP-process  $P_2$  of Section 3. In the constructed TT-process  $T_2$  we could have taken the same symbol for  $c_{|a|}$  and  $c_{|b|}$  (but different from  $C_0$ ).

To clarify the differences between P1a, P1b, and P1c, we present the state graphs of the TT-processes constructed for the CSP-process  $P_0$  of Section 1. They are called  $Ta_0$ ,  $Tb_0$ , and  $Tc_0$  respectively.



The second stage also allows some reductions. An alternative definition of  $T$  given  $S$  is

$$T = \langle \mathbf{a}T \cup \{c\} \\ , \{t \in \mathbf{t}S \mid t \upharpoonright B \notin E\} \cup \{tc^n \mid t \in \min(S, B, E) \wedge n \geq 0\} \\ \rangle$$

where  $c$  is a fresh symbol if  $\mathbf{a}S = B$ , otherwise  $c \in \mathbf{a}S \setminus B$ ; and  $\min(S, B, E)$  is the set of minimal traces in  $\mathbf{t}S$  (with respect to the prefix order) for which the projection on  $B$  is in  $E$ , that is

$$\min(S, B, E) = \{t \in \mathbf{t}S \mid (\mathbf{A} u : u \leq t \wedge u \upharpoonright B \in E : t = u)\}$$

This definition of  $T$  relies for its correctness on the fact that  $D_B$  and  $F_B$  in H2 satisfy C5 and C6, no matter what  $T$  is projected. For instance, the TT-process  $U_4$  was constructed with this new definition.

## 5 STATE GRAPHS

A convenient representation of a TT-process is its state graph. This representation is based on the following equivalence relation.

Let  $T$  be a TT-process. Traces  $t$  and  $u$ ,  $t \in \mathbf{t}T$  and  $u \in \mathbf{t}T$ , belong to the same *state* of  $T$ , denoted by  $t \mathbf{e}_T u$ , when

$$(\mathbf{A} v : v \in (\mathbf{a}T)^* : tv \in \mathbf{t}T \equiv uv \in \mathbf{t}T)$$

Many definitions and properties of TT-processes can also be expressed in terms of their states, i.e., sets of equivalent traces instead of individual traces. A similar relation is possible for CSP-processes.

Let  $P$  be a CSP-process. Traces  $t$  and  $u$ ,  $t \in \mathbf{t}P$  and  $u \in \mathbf{t}P$ , belong to the same *state* of  $P$ , denoted by  $t \mathbf{e}_P u$ , when

$$(\mathbf{A} v : v \in (\mathbf{a}P)^* : f(tv) = f(uv))$$

where the function  $f$  maps traces over  $aP$  on pairs consisting of a subset of  $Powerset(aP)$  and a Boolean, as defined by

$$f(t) = \langle \{X \mid (t, X) \in fP\}, (t \in dP) \rangle$$

Notice that  $t \in tP$  is equivalent to  $\{X \mid (t, X) \in fP\} \neq \emptyset$  on account of C2.

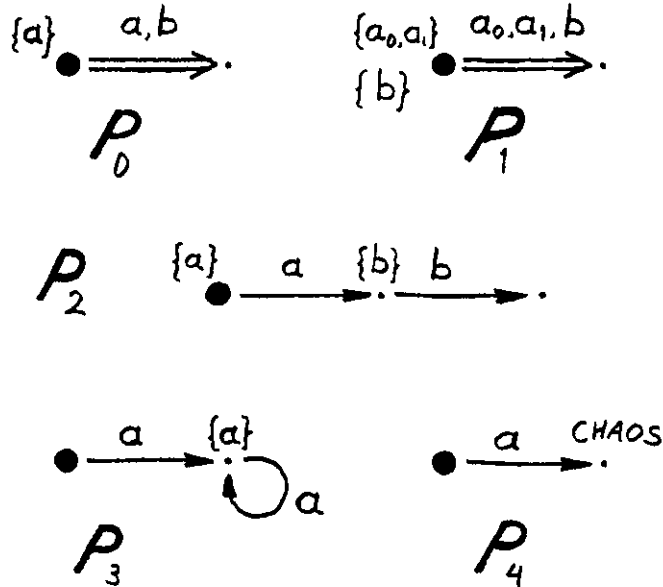
For traces  $t$  and  $u$  in  $tP$  with  $t e_P u$  we have

$$S(t, P) = S(u, P) \wedge M(t, P) = M(u, P)$$

In the state graph for CSP-process  $P$  we, therefore, label the state containing trace  $t$ ,  $t \in tP$ , with the refusal sets  $X$  in  $M(t, P)$ . On account of C5 and C6  $dP$ —if nonempty—forms one state of  $P$ ; in the state graph of  $P$  this state is labeled *CHAOS*, and all outgoing arrows and refusal sets are omitted.

A CSP-process is deterministic if and only if its state graph has no labeled states. If it is unlabeled its state graph is also the state graph of the corresponding TT-process.

The state graphs of the CSP-processes  $P_0$  (Section 1), and  $P_1, P_2, P_3$ , and  $P_4$  (Section 3) are given to illustrate the notation.

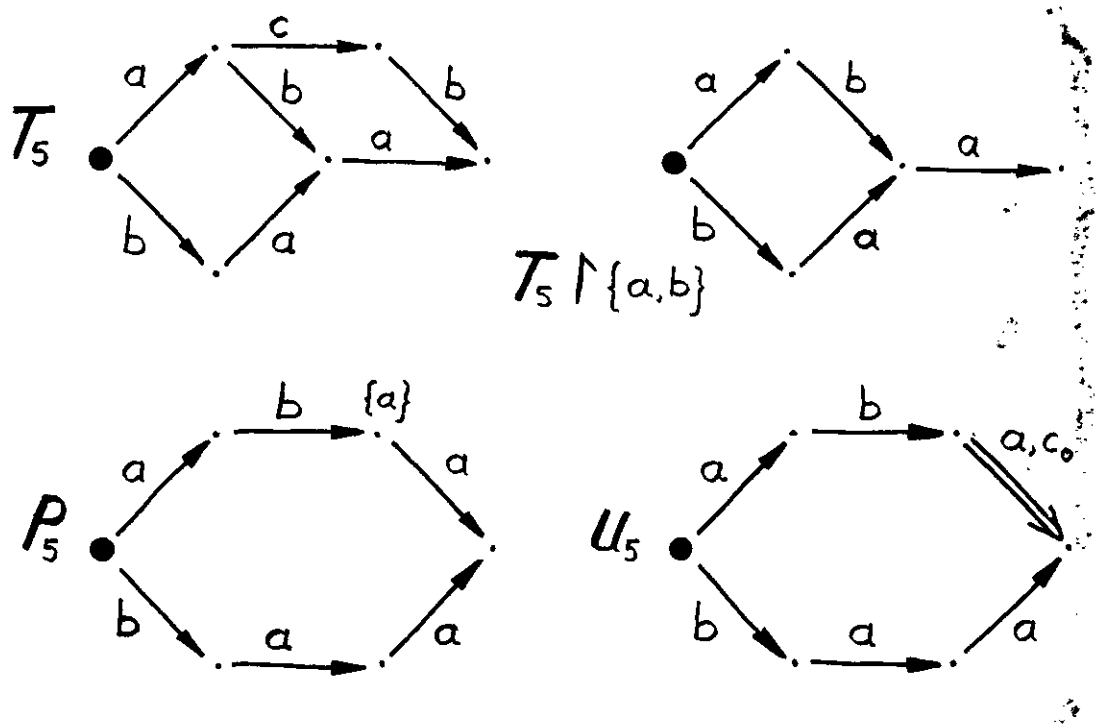


For traces  $t$  and  $u$ ,  $t \in tP$  and  $u \in tP$ , we have

$$t e_P u \Rightarrow t e_{tr(P)} u$$

The converse does not hold. Consider the TT-process  $T_5$  given by the state graph below. Let  $P_5$  be the CSP-process  $pr(T_5) \upharpoonright \{a, b\}$ . Here we have  $T_5 = tr(P_5)$ , but

$$ab e_{T_5} ba \wedge \neg(ab e_{P_5} ba)$$



For TT-process  $T$  and traces  $t$  and  $u$ ,  $t \in \mathbf{t}T$  and  $u \in \mathbf{t}T$ , we do have

$$t \mathbf{e}_T u \equiv t \mathbf{e}_{pr(T)} u$$

We now describe how the state graph of CSP-process  $Q$  can be transformed into the state graph of TT-process  $T$  for which  $pr(T) \upharpoonright \mathbf{a}Q = Q$ .

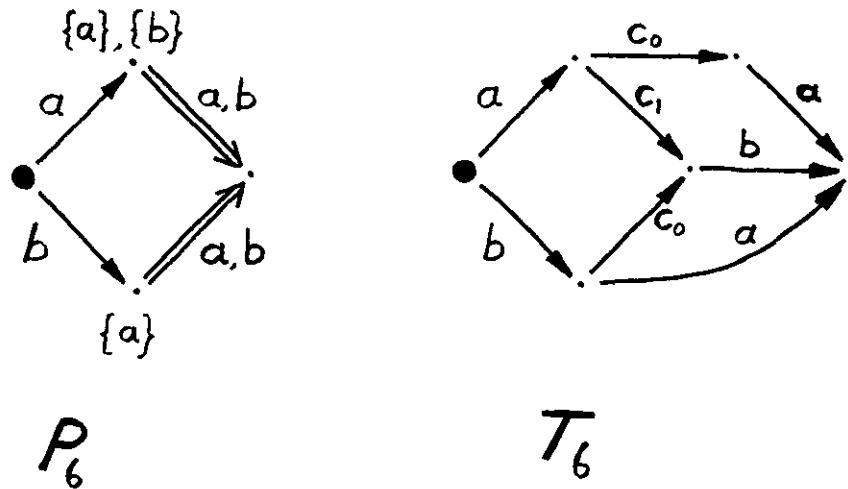
Let  $n$  be the maximum number of labels that occur on states of  $Q$ . Let  $c_i$ ,  $0 \leq i < n$ , be a fresh symbol, that is  $\{c_i \mid 0 \leq i < n\} \cap \mathbf{a}Q = \emptyset$ . Add an arrow labeled  $c_0$  from state  $CHAOS$  to state  $CHAOS$  and omit the label  $CHAOS$  from the state graph.

For state  $q$  with labels  $X_i$ ,  $0 \leq i < k$ : add  $k$  new vertices  $q_i$ ; for each  $i$ ,  $0 \leq i < k$ , and each arrow labeled  $b$ ,  $b \notin X_i$ , from state  $q$  to state  $r$ , mark that arrow and add a new arrow labeled  $b$  from vertex  $q_i$  to state  $r$ ; remove all marked arrows leaving state  $q$ ; for each  $i$ ,  $0 \leq i < k$ , add a new arrow labeled  $c_i$  from state  $q$  to vertex  $q_i$ ; remove all labels from state  $q$ .

The resulting graph need not be a state graph. The problem is that different vertices may belong to the same state, that is, the graph need not be minimal. For each particular state  $q$  the new vertices  $q_i$  are in distinct states, because their successor sets differ. But a vertex  $q_i$  may belong to some existing state or to the same state as one of the other new vertices. After minimization the resulting graph is the state graph of a deterministic CSP-process whose projection equals  $Q$ .

If this transformation is applied to the state graph of  $P_5$  above one obtains the state graph for  $U_5$  depicted above. Here, a new vertex is swallowed by an existing state during minimization.

As a final example we give the state graph of CSP-process  $P_6$  and the transformed graph after minimization. Two new vertices were merged.



## 6 CONCLUSION

We have shown that for each CSP-process  $Q$  there exists a *deterministic* CSP-process  $P$  such that  $P \upharpoonright a Q = Q$ . The proof we have presented is constructive and has also been explained in terms of state graphs.

CSP can be viewed as an extension or refinement of Trace Theory. CSP attempts to capture deadlock and livelock phenomena under all "reasonable" operators. In Trace Theory projection is always considered a problematic operator. The main result of this paper shows that CSP is a *minimal* extension of Trace Theory covering the projection operator: All distinctions that CSP makes between processes can be justified by projecting suitable deterministic processes—that is, processes in the sense of Trace Theory. This also means that all problems with deadlock and livelock in Trace Theory can be attributed to the projection operator—insofar these problems can be solved within CSP, of course.

Acknowledgment is due to Martin Rem and Anne Kaldewaij for their comments on a draft version of this paper.

## APPENDIX

Let  $T$  be a TT-process and let  $B$  be a subset of  $\mathbf{a}T$ . The alphabet  $B$  is called *transparent* with respect to  $T$  (see Section 5.2 of [1]) when

$$\begin{aligned} & (\mathbf{A} t : t \in \mathbf{t}T : S(t, T) \subseteq B \Rightarrow S(t, T) = S(t \upharpoonright B, T \upharpoonright B)) \wedge \\ & (\mathbf{A} t : t \in \mathbf{t}T : (\mathbf{E} n : n \geq 0 : (\mathbf{A} s : s \in (\mathbf{a}T \setminus B)^* \wedge ts \in \mathbf{t}T : l(s) \leq n))) \end{aligned}$$

The second conjunct of this definition is abbreviated *livelockfree*( $\mathbf{a}T \setminus B, T$ ). We prove the **Transparency Theorem**

$$B \text{ is transparent w.r.t. } T \equiv pr(T) \upharpoonright B = pr(T \upharpoonright B)$$

**Proof** Let us write  $P$  for  $pr(T)$ , hence  $P$  is deterministic and  $tr(P) = T$ . We derive

$$\begin{aligned} & B \text{ is transparent w.r.t. } T \\ \equiv & \quad \{ \text{def. of transparent w.r.t. CSP-process, using } P = pr(T) \text{ and } tr(P) = T \} \\ & B \text{ is transparent w.r.t. } P \\ \equiv & \quad \{ \text{Theorem 5.3.8 in [1]} \} \\ & P \upharpoonright B \text{ is deterministic} \\ \equiv & \quad \{ \text{DET} \} \\ & P \upharpoonright B = pr(tr(P \upharpoonright B)) \\ \equiv & \quad \{ \text{definition of } pr \text{ and calculus} \} \\ & P \upharpoonright B = pr(tr(P \upharpoonright B)) \wedge \mathbf{d}(P \upharpoonright B) = \emptyset \\ \equiv & \quad \{ \text{Property 5.3.7 in [1], using that } P \text{ is deterministic} \} \\ & P \upharpoonright B = pr(tr(P \upharpoonright B)) \wedge \text{livelockfree}(\mathbf{a}T \setminus B) \\ \equiv & \quad \{ \text{definition of } tr, \text{ set calculus, and Property 5.3.6 in [1]} \} \\ & P \upharpoonright B = pr(tr(P) \upharpoonright B) \wedge \text{livelockfree}(\mathbf{a}T \setminus B) \\ \equiv & \quad \{ \text{Property 5.3.7 of [1], using that } P \text{ is deterministic} \} \\ & P \upharpoonright B = pr(tr(P) \upharpoonright B) \wedge \mathbf{d}(P \upharpoonright B) = \emptyset \\ \equiv & \quad \{ \text{definition of } pr \text{ and calculus} \} \\ & P \upharpoonright B = pr(tr(P) \upharpoonright B) \\ \equiv & \quad \{ P = pr(T) \text{ and } tr(P) = T \} \\ & pr(T) \upharpoonright B = pr(T \upharpoonright B) \end{aligned}$$

(End of Proof)

N.B. Theorem 5.3.8 in [1] is equivalent to the Transparency Theorem.

## REFERENCES

- [0] C.A.R. Hoare, *Communicating Sequential Processes*, Prentice-Hall, 1985.
- [1] A. Kaldewaij, *A Formalism for Concurrent Processes*, Dissertation, Eindhoven University of Technology, Eindhoven, The Netherlands, May 1986.

## COMPUTING SCIENCE NOTES

In this series appeared :

No.	Author(s)	Title
85/01	R.H. Mak	The formal specification and derivation of CMOS-circuits
85/02	W.M.C.J. van Overveld	On arithmetic operations with M-out-of-N-codes
85/03	W.J.M. Lemmens	Use of a computer for evaluation of flow films
85/04	T. Verhoeff H.M.J.L. Schols	Delay insensitive directed trace structures satisfy the foam rubber wrapper postulate
86/01	R. Koymans	Specifying message passing and real-time systems
86/02	G.A. Bussing K.M. van Hee M. Voorhoeve	ELISA, A language for formal specifications of information systems
86/03	Rob Hoogerwoord	Some reflections on the implementation of trace structures
86/04	G.J. Houben J. Paredaens K.M. van Hee	The partition of an information system in several parallel systems
86/05	Jan L.G. Dietz Kees M. van Hee	A framework for the conceptual modeling of discrete dynamic systems
86/06	Tom Verhoeff	Nondeterminism and divergence created by concealment in CSP
86/07	R. Gerth L. Shira	On proving communication closedness of distributed layers

86/08	R. Koymans R.K. Shyamasundar W.P. de Roever R. Gerth S. Arun Kumar	Compositional semantics for real-time distributed computing (Inf.&Control 1987)
86/09	C. Huizing R. Gerth W.P. de Roever	Full abstraction of a real-time denotational semantics for an OCCAM-like language
86/10	J. Hooman	A compositional proof theory for real-time distributed message passing
86/11	W.P. de Roever	Questions to Robin Milner - A responder's commentary (IFIP86)
86/12	A. Boucher R. Gerth	A timed failure semantics for communicating processes
86/13	R. Gerth W.P. de Roever	Proving monitors revisited: a first step towards verifying object oriented systems (Fund. Informatica IX-4)
86/14	R. Koymans	Specifying passing systems requires extending temporal logic
87/01	R. Gerth	On the existence of sound and complete axiomatizations of the monitor concept
87/02	Simon J. Klaver Chris F.M. Verberne	Federatieve Databases
87/03	G.J. Houben J.Paredaens	A formal approach to distri- buted information systems
87/04	T.Verhoeff	Delay-insensitive codes - An overview