CIS 525 Parallel and Distributed Software Development

Algorithm Computing Synchronic Distance of two sets of events E_1 and E_2 of C/E nets

Given $\sum=(B_\sum, E_\sum, F_\sum, C_\sum)$ contact-free C/E system compute synchronic distance between two sets of events.

Step 1: For $\sum$ compute $\prod_\sum$ - the set of all finite processes of $\sum$; each process will be represented by a pair $(K_\sum, p)$, where $K_\sum$ is an occurrence net and p is a mapping $p: S \rightarrow B_\sum, T \rightarrow E_\sum$

Step 2: For each occurrence net calculated in step 1 compute pre-images of events E_1 and E_2, $p^{-1}(E_1)$ and $p^{-1}(E_2)$.

Step 3: For each occurrence net calculated in step 1 compute a set of all slices $sl(K_\sum) = \{D_1,D_2,\ldots,D_m\}$.

Step 4: For each pre-image of E_1 and E_2 (as computed in step 2) assign μ - a measure of distance between arbitrary two slices in $K_\sum$

$$\mu(p^{-1}(E_i), D_j, D_k) = \begin{cases} |p^{-1}(E_i) \cap D_j^{+} \cap D_k^{-}| - |p^{-1}(E_i) \cap D_j^{-} \cap D_k^{+}| & \text{for incomparable slices} \\ |p^{-1}(Ei) \cap D_j^{+} \cap D_k^{-}| & \text{if } D_j < D_k \\ |p^{-1}(E_i) \cap D_j^{-} \cap D_k^{+}| & \text{if } D_k < D_j \end{cases}$$

$i = 1,2$

$j, k = 1,2,..,m$

Step 5: Compute a variance $v(p, E_1, E_2)$ between events E_1 and E_2 process $(p: K_\sum \rightarrow \sum), \in \prod_\sum$

$$v(p,E_1,E_2) = \max\{ \mu(p^{-1}(E_1), D_j, D_k) - \mu(p^{-1}(E_2), D_j, D_k) : D_j,D_k \in sl(K)\}$$

Step 6: Compute a synchronic distance between sets of events E_1 and E_2

$$\sigma(E_1, E_2) = \sup \{v(p,E_1,E_2): p \in \textstyle\prod_\sum\}.$$

How to compute a set of all finite processes for a given C/E system?

There is a theorem which states that for each path of a case graph there is exactly one corresponding process. So, using this theorem one can calculate all possible paths in the case graph, and then to compute a process for this path. This is possible because of uniqueness of this relationship.

SYNCHRONIC DISTANCE – EXAMPLES

1: $\sum_1$ $\sigma(E_1, E_2)$

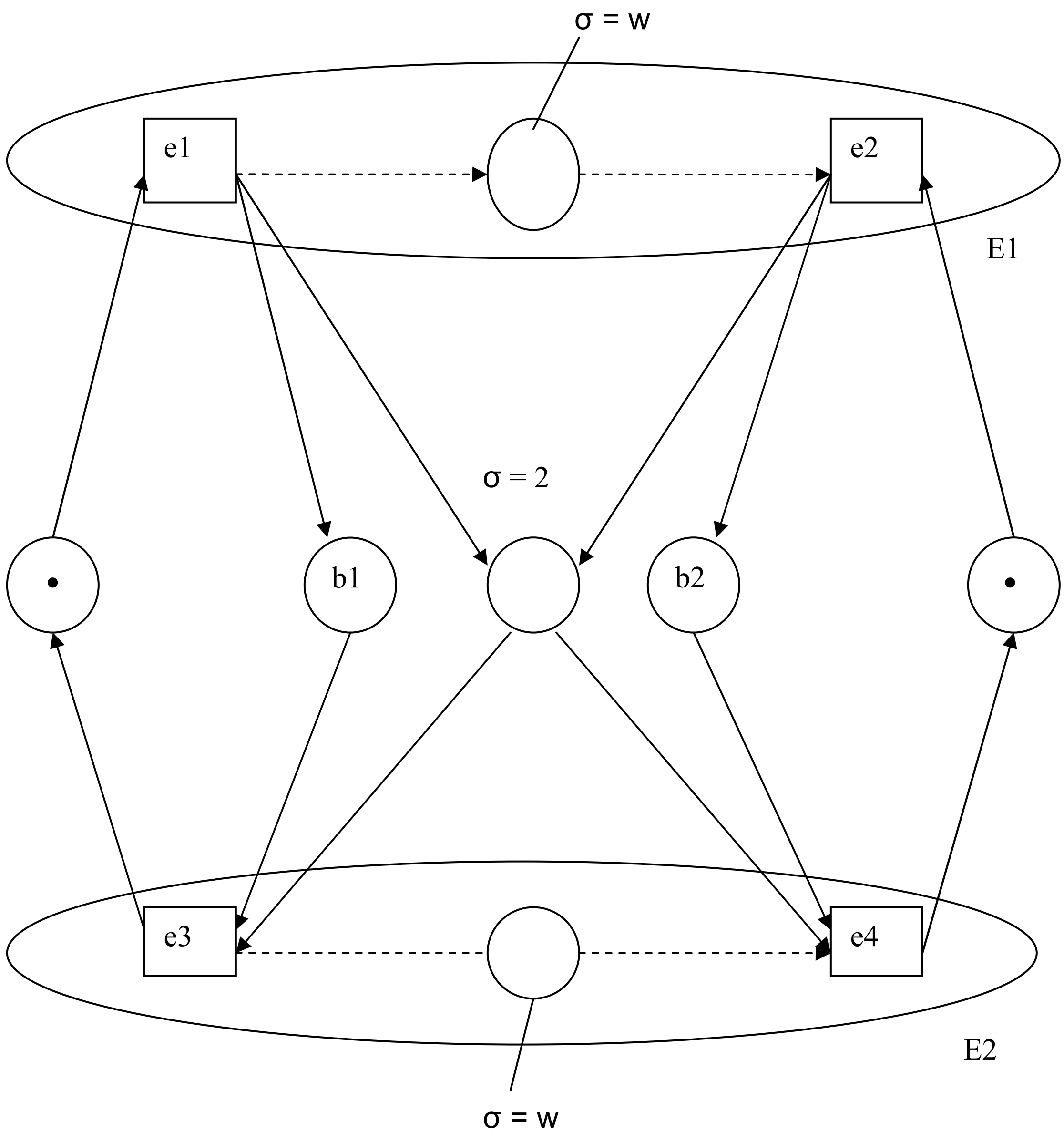

Figure 1. Illustration of the concept of synchronic distance.

$\sigma(E_1, E_2) = 2$

$\sigma(e_1, e_4) = \sigma(e_2, e_3) = \sigma(e_1, e_2) = \sigma(e_3, e_4) = w$

$\sigma(e_1, e_3) = \sigma(e_2, e_4) = 1$

2: $\sum_2$

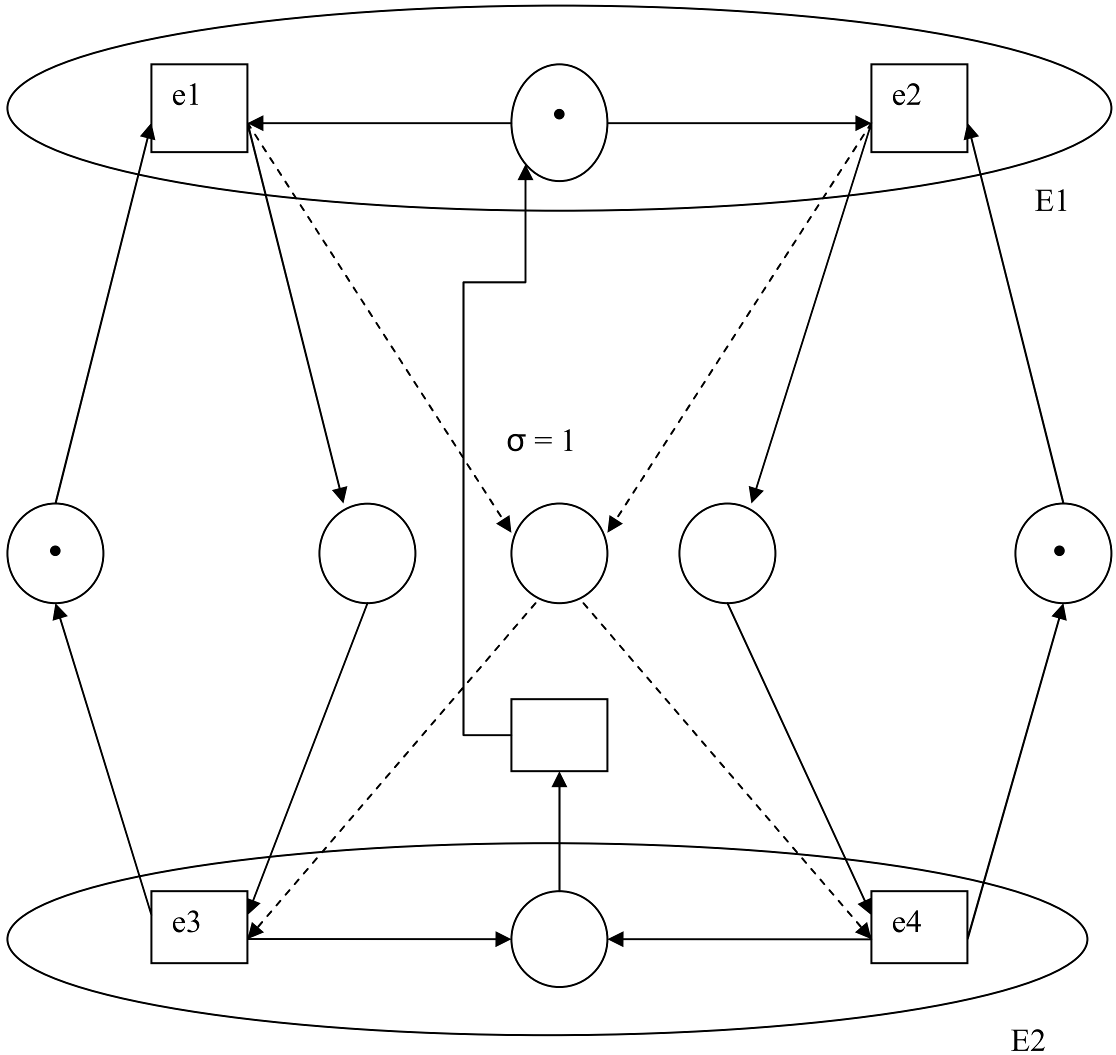

Figure 2. Illustration of the concept of synchronic distance.

$\sigma(\{e_1,e_2\},\{e_3,e_4\}) = 1$

<u>Remark</u>

Because of these two results we consider $\sum_2$ as more strictly synchronized.

3: Example with weighted synchronic distance:

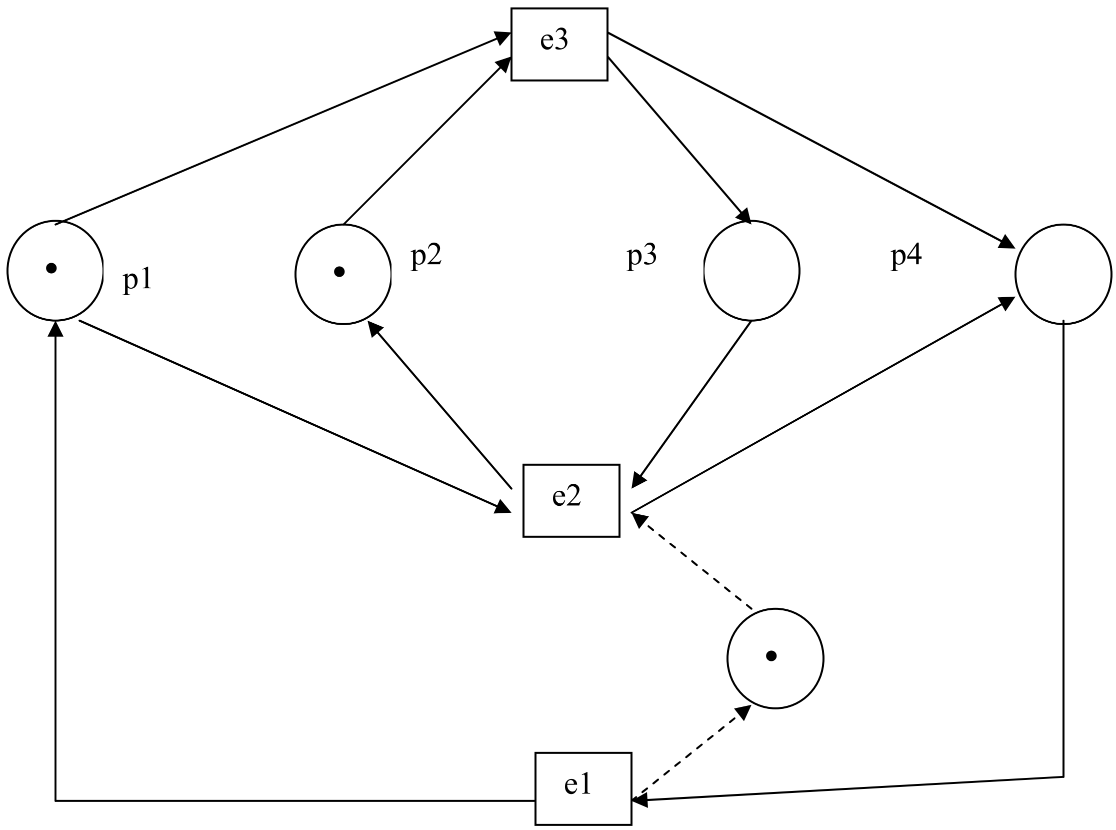

Figure 3. Illustration of the concept of weighted synchronic distance.

Case graph

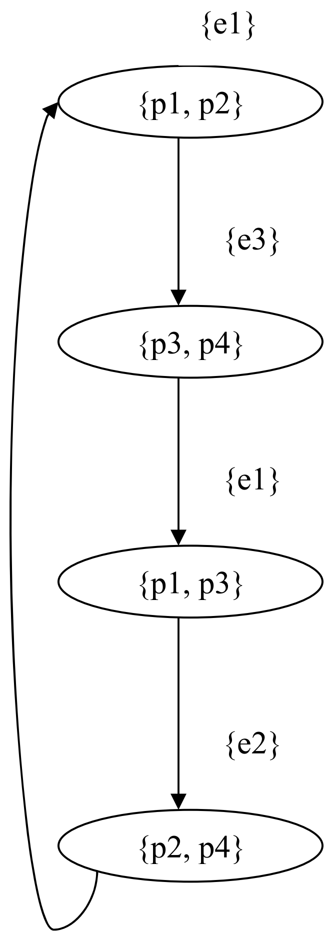

Figure 4. Case graph for a C/E net.

<u>Example</u>: Implement a system by specifying necessary synchronic substances. Let us assume that the producer and consumer have agreed that the capacity of their shared store house should be 4.

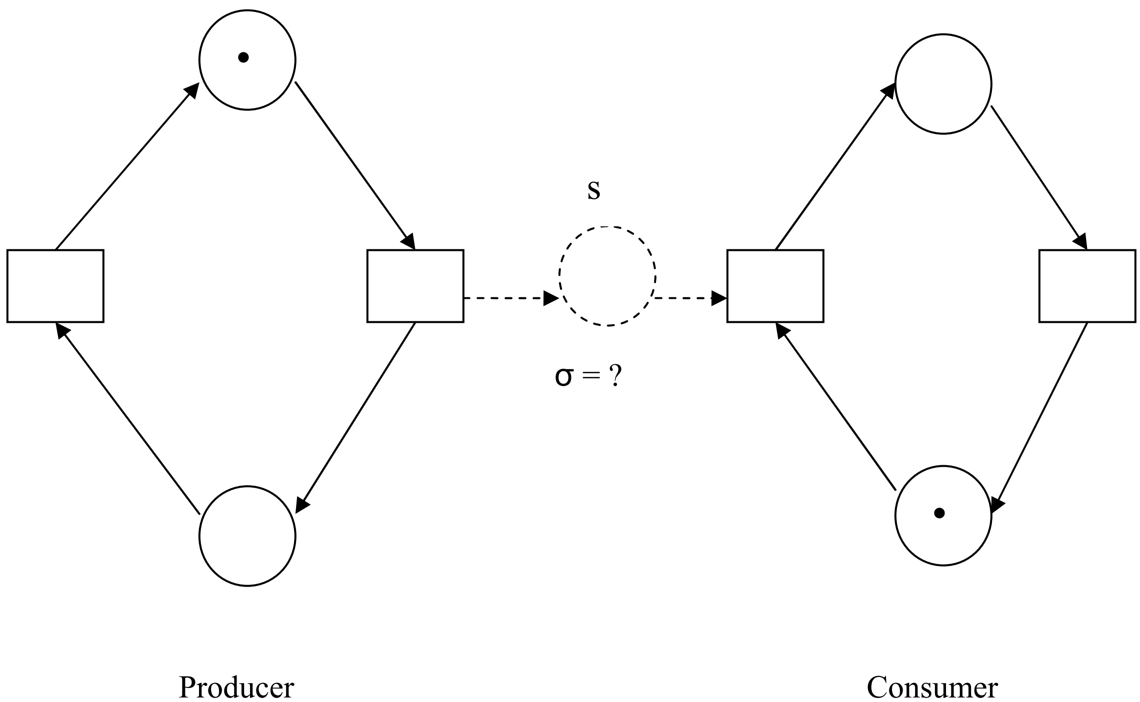

Figure 5. Synchronic distance of the Producer-Consumer system.

Implementation:

1.

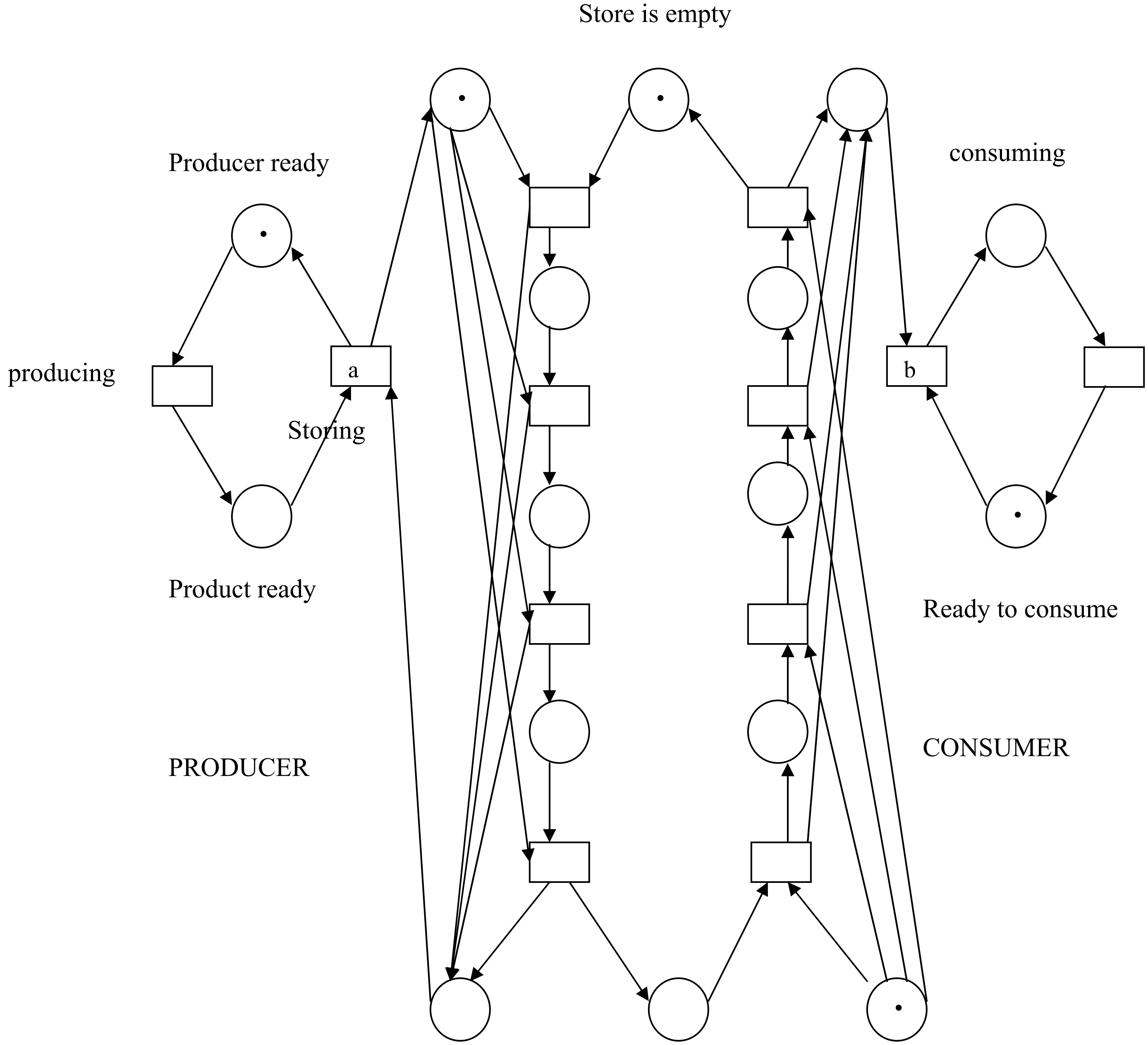

Figure 6. Producer-consumer model with synchronic distance.

Remark: This solution allows no access from the consumer until the house is full, and no access from the producer until the house is empty.

2.

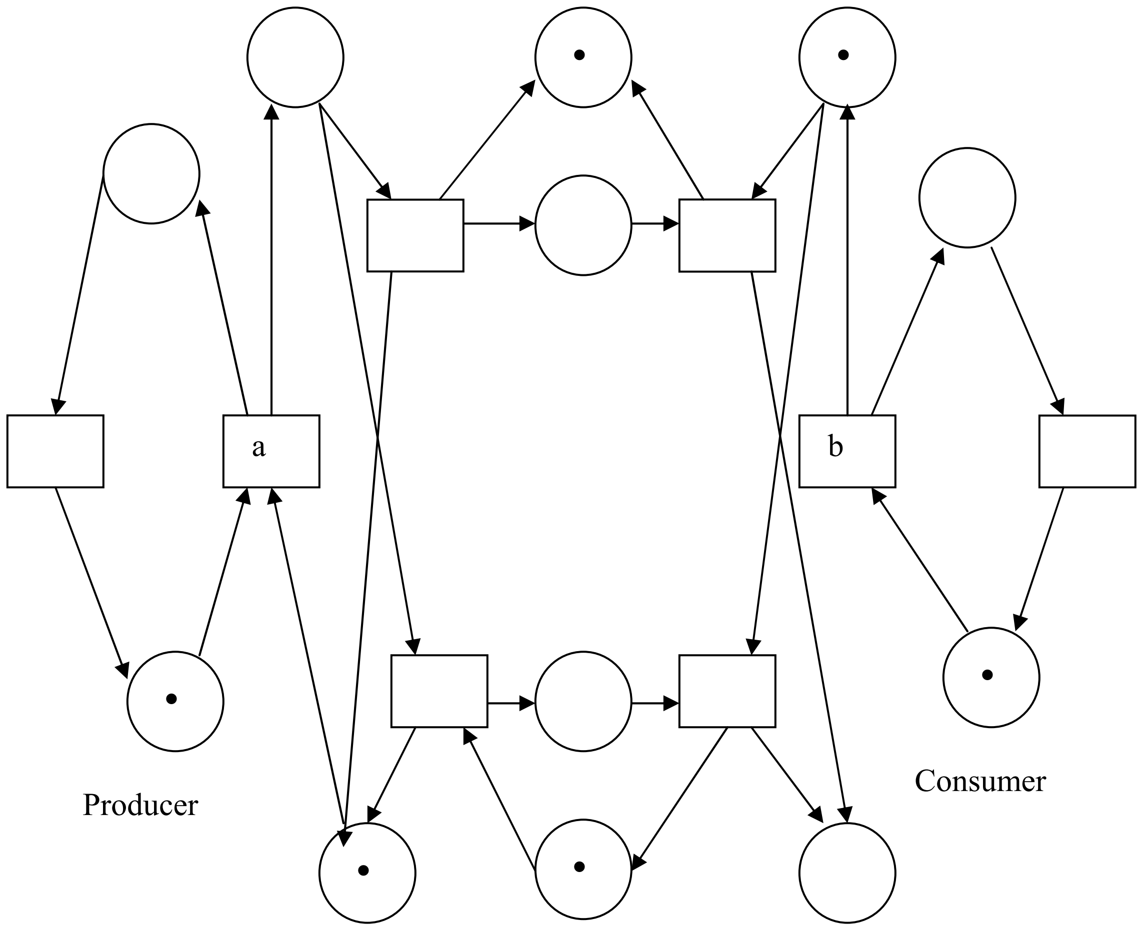

Figure 7. Producer-Consumer system with synchronic distance.

Synchronic distance of cyclic and non-cyclic systems.

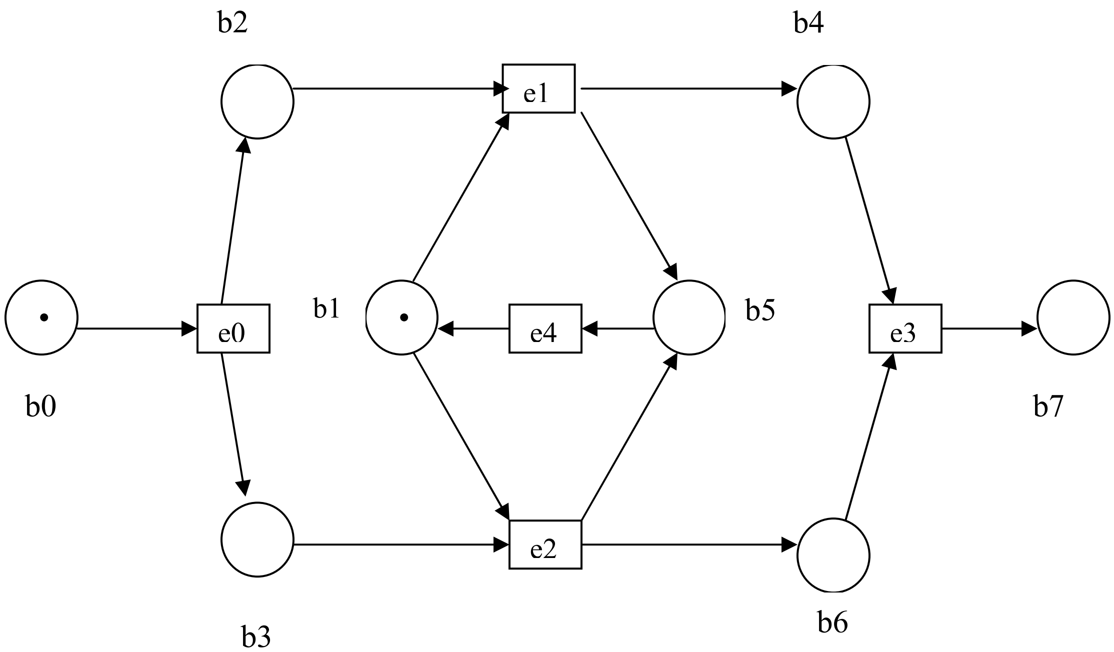

Figure 8. Synchronic distance for non-cyclic system.

This system is not cyclic because its case graph is not strongly connected.

Case graph

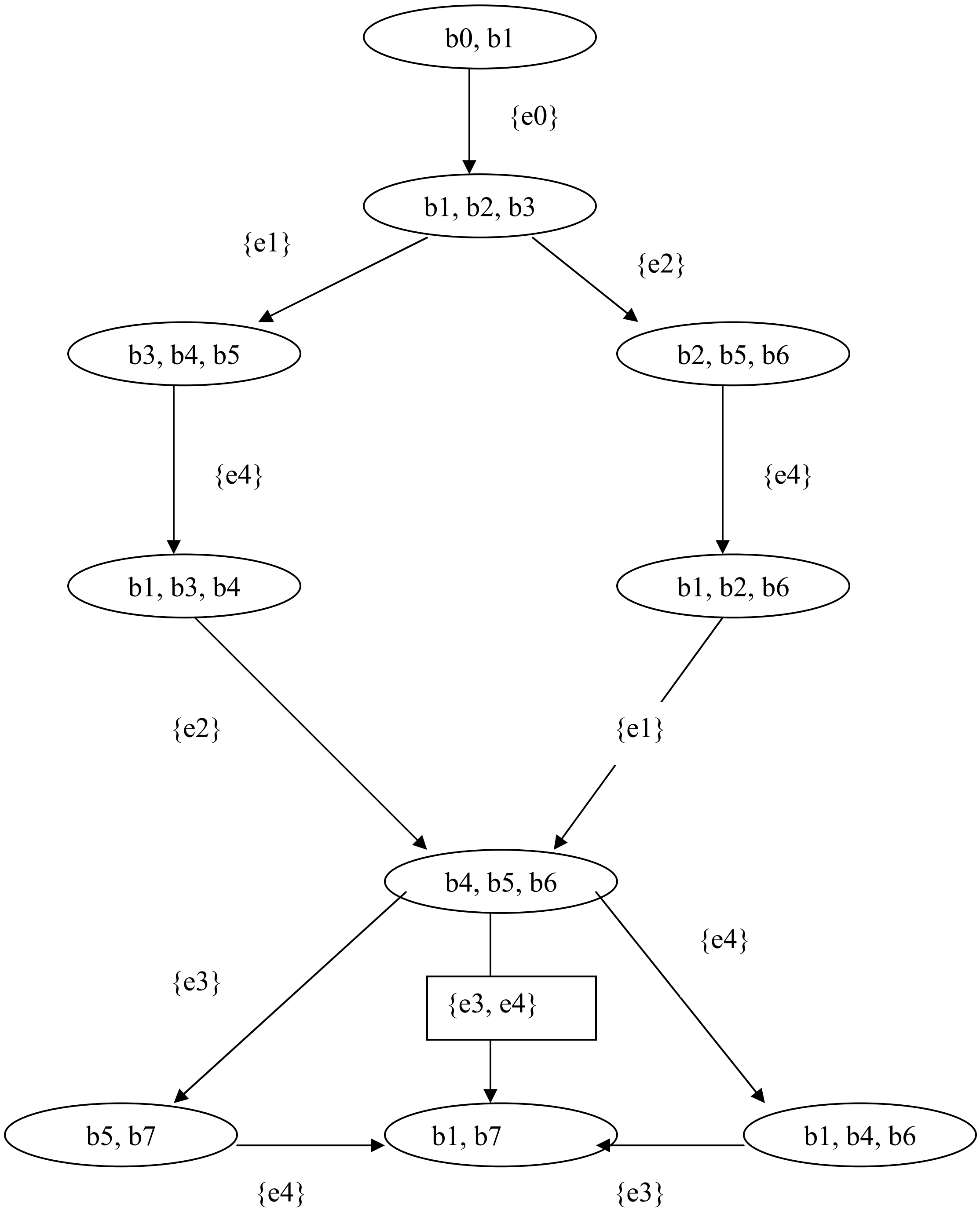

Figure 9. Case graph

Remark1: $a = e_1$, $b = e_2$ Events a and b can still occur in any order and therefore there is no difference between system with b_1, b_5, and e_4 and between system without this element (as long as e_1 and e_2 is concerned). But events e_1, e_2 were previously concurrent and now are in conflict. One can verify that now $\sigma(e_1,e_2) = 1$.

Remark 2: Sometimes it is necessary to distinguish concurrency from sequential (perhaps no deterministic) behavior. Synchronic distance may sometimes be used to distinguish concurrency from arbitrary interleaving. If an event a may occur concurrently with some event b, then we will have $\sigma(a, b) \geq 2$ while we might have $\sigma(a, b) = 1$ in the case of arbitrary interleaving.

Example: How to decide whether or not a weight function exists such that a finite synchronic distance may be obtained between two sets of events and how to find such a weight function?

Theorem. Let $\sum = (B, E, F, C)$ be a C/E system and $E_1, E_2 \subseteq E_1$ then we have

$$\sigma(E1, E2) = \begin{cases} w & \text{if } \exists p \in RP(\Sigma) : |p^{-1}(E_1)| - |p^{-1}(E_2)| \neq 0 \\ \max \{v(p, E1, E2) \mid p \in SP(\Sigma)\} & \text{otherwise} \end{cases}$$

where $RP(\Sigma)$ and $SP(\Sigma)$ are the set of <u>reproduction processes</u> and the set of <u>simple processes</u> of Σ.

Def. Le $p \in RP(\Sigma)$ be a process, p is cyclic iff $p(^{\circ}K) = p(K^{\circ})$ where K is the occurrence net of p.

Def. Let $p, p' \in PR(\Sigma)$. p' is called a <u>proper subprocess</u> of p iff $p_1, p_2 \in PR(\Sigma)$ such that $p = p_1 \circ p' \circ p_2$ and p_1 or p_2 is not empty.

Def. Processes without proper cyclic subprocesses are called <u>simple</u>. A process is a <u>reproduction process</u> if it is cyclic and simple.

Example: C/E system

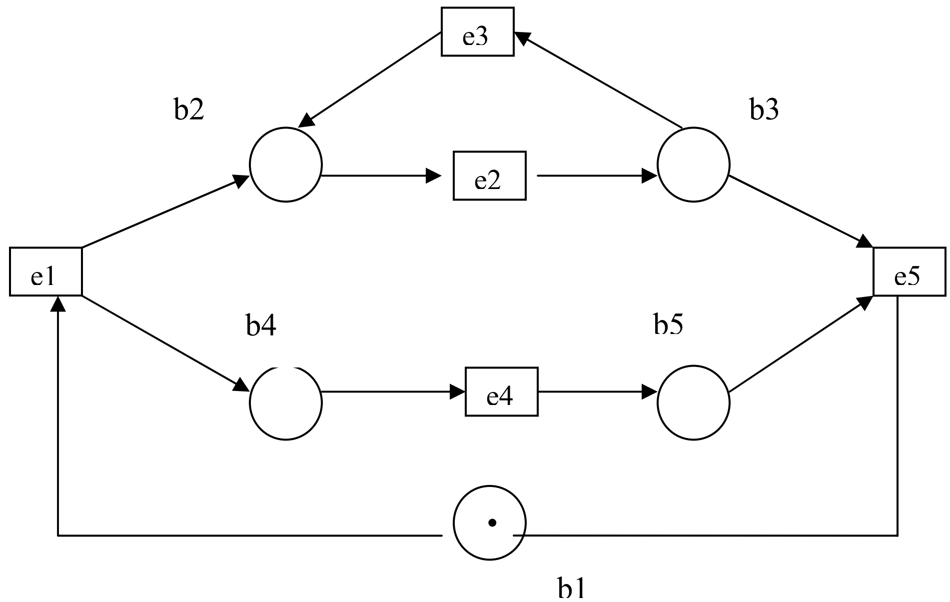

Figure 10. C/E net.

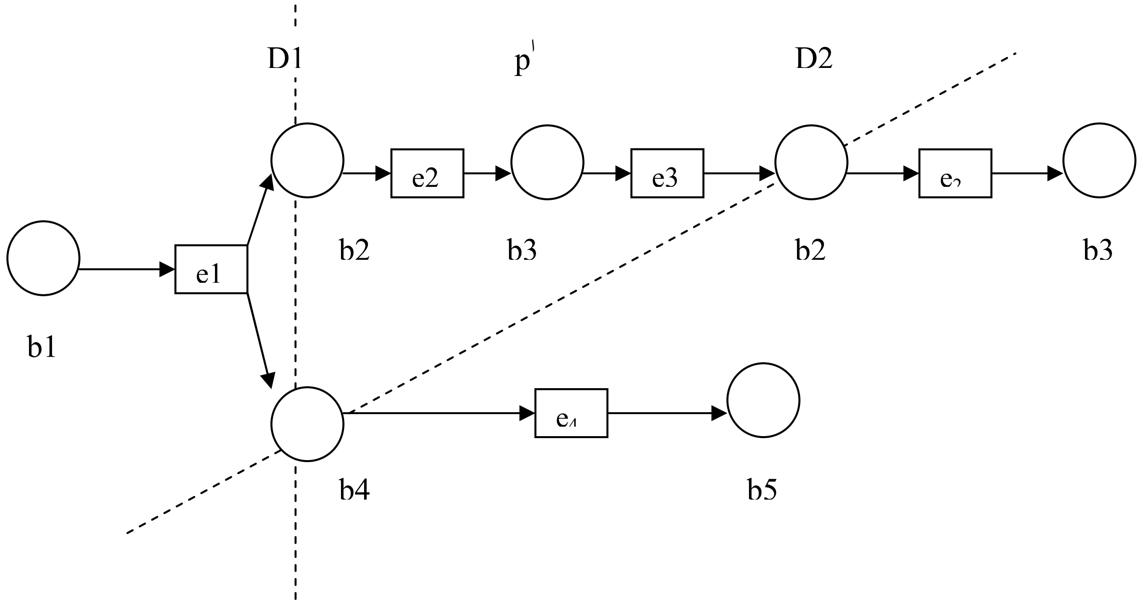

Figure 11. Process net.

1) Process is not simple

2) Proper subprocess between slices D_1 and D_2, denoted as p'; this is reproduction process.

Remark: Using this theorem we may consider only reproduction processes. These determine a linear equation system such that its solutions are weight functions which yield a finite synchronic distance.

Problem: We would like to find weights for finite synchronic distance in all cases where the sets of events E_1 and E_2 are synchronized in the following way:

There exists $n \in N$ such that we may not have more than n occurrences of events in E_1 without intermediate occurrence of some event in E_2 or vice versa. Unfortunately, this is not always possible. Example of this is presented on p. 59. In this system we have two reproduction processes. In the one operating in the left part of the system, one occurrence of e_0 is followed by two occurrences of e_1. Conversely, we have two occurrences of e_0 and one occurrence of e_1 in the other reproduction process. Hence there is no way to weight e_0 and e_1 to obtain a finite synchronic distance even though we never have more than two occurrences of any event in sequence.