Bulletin of EATCS, No. 37 Feb. 1989

A Look at Concurrency Semantics through "Lattice Glasses"

Leonid Rosenblum, Alexandre Yakovlev
Vladimir Yakovlev*

Leningrad Electrical Engineering Institute
Professor Popov Street 5
Leningrad 197022 U.S.S.R.

Recently, a series of attempts has been undertaken to tackle the controversy between causal (partial order) semantics and interleaving semantics /1, 2, 3/.

The main result of these exciting discussions can be summarized as follows: although the interleaving semantics can be very suitable in certain (especially, simulation-oriented) circumstances, the causal view upon concurrency is generally more versatile, especially when one has to deal with refinable specifications /1/, or when specific properties like confusion, choice-absence, strong concurrency and "causal-next" relation, inherent in causal semantics /3/, should be treated.

◇It is interesting to look at concurrent behaviour with more attention using some algebraic classification criteria. To make our discussion more clear and easy-comprehensible we restrict ourselves with processes that are free from alternatives in their execution. Strictly speaking, we allow only those alternative execution sequences which are due to concurrency and its "non-deterministic" simulation, but not because of choices or conflict resolutions. Again, for the sake of clarity, we base our observations on

Example.

Consider two Petri nets, PN1 and PN2, as shown in Fig.1. The corresponding marking diagrams, MD1 and MD2, for these nets are shown in Fig.2.

*) In 1988, a visitor to UMIST, Manchester, England.

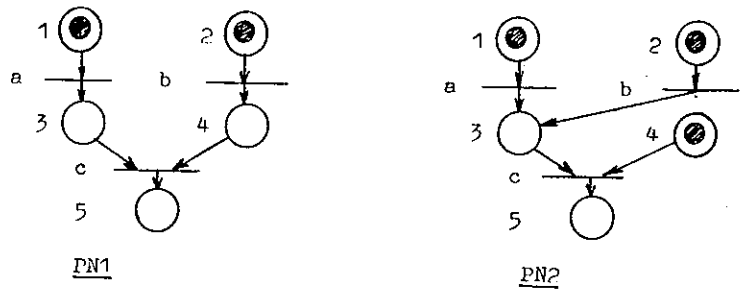


Fig.1

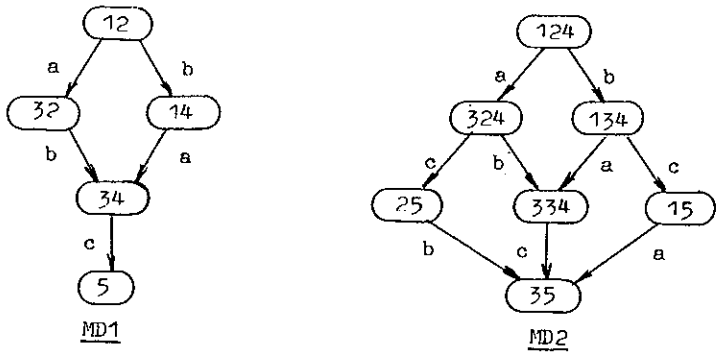


Fig.2

The execution sequences, or trace sets, for these nets are given by ES1 = { abc, bac } and ES2 = { abc, bac, acb, bca }, respectively.

We also consider the so-called cumulative diagrams which can be built on the sets of firing vectors for our two processes. Such vectors are composed of current values of numbers of firings for every action of a process (of course, with respect to a given initial marking). For PN1 and PN2, the cumulative diagrams CD1 and CD2 are shown in Fig.3.

◇I. It is easily seen that such diagrams are the graphical representations (Hasse diagrams) of partially ordered sets of firing vectors (cumulative states).

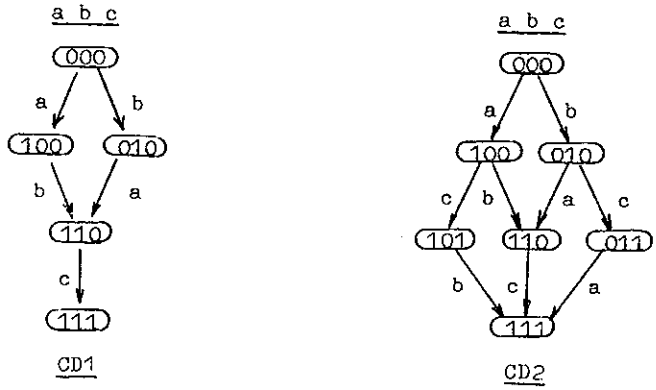


Fig.3

From these two posets one can deduce that for PN1 $P01 = (\{000, 100, 010, 110, 111\}, \leq)$ forms a distributive lattice with the zero element 000, and for PN2 $P02 = (\{000, 100, 010, 101, 110, 011, 111\}, \leq)$ forms a semimodular lattice with 000 zero. The latter is not distributive that can be shown from the absence of the 001 combination in the lattice.

(Since for $\alpha = 101$ and $\beta = 011$ $\delta = \alpha \wedge \beta$ is given by $\delta_i = \min(\alpha_i, \beta_i)$, $i = 1,2,3$, then $\delta = 001$, whereas the greatest lower bound $\alpha \cap \beta$ is 000, and, therefore, $\exists \alpha, \beta : \alpha \wedge \beta \neq \alpha \cap \beta$, thus proving that the P02 lattice is not distributive.)

Similar observations had been made more than 25 years ago by D.E.Muller in /4/ in his study of subclasses of speed-independent circuits. Muller had shown that for distributive circuits (i.e. those that generate distributive cumulative diagrams of logical element switchings), one can equally use both partial order semantics on actions (he considered the binary signal changes) and operational semantics given by the lattice of cumulative states. Hence, only distributive processes are able to preserve the full information of causal dependence between actions in any

of the two counterpart representations (see also /5/).

Thus, for PN1 one can use the knowledge of ES1 to reconstruct the partial order of the form $a < c$ and $b < c$. But for PN2, there is no partial order between a, b and c.

Note. W.Reisig /3/, however, extends the notation of partial order, to cover processes like PN2, by introducing the ordering of the type

" $a \xrightarrow{b} c$ where $a < c$ and b is unordered with both a and c ,

and (*)

$b \xrightarrow{a} c$ where $b < c$ and a is unodered with both b and c".

This notation seems, however, not very successful for the described situation because from the logical point of view using the "and" marked with (*) is rather confusing in such circumstances, as it leads to the contradiction between, say, having "$a < c$" in the first clause and " a is unordered with ... c" in the second clause.

Hence, the following observation can obviously be stated.

Observation 1. For "distributive" form of concurrency the interleaving semantics and causal semantics are equally informative.

One should, of course, bear in mind that by this we do not lift any complaints concerning the inefficiency of the interleaving semantics with respect to refinability issues /1/.

◇II. Another interesting observation about distributive concurrency is attributable to the relational view upon concurrency.

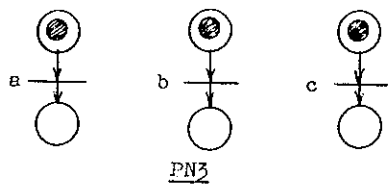
Let us consider the $\|$-relation which can be defined between pairs of events in a given process. Let for events a and b the $\|$-relation be defined as

$$a \parallel b = \neg(a \prec b) \wedge \neg(b \prec a) =$$
$$= \neg(a \prec b \vee b \prec a)$$

where the $a \prec b$ relation can be in its turn obtained from the interleaving semantics given by the set of execution sequences as the fact that a always precedes b in every execution sequence.

From this definition we may see that for PN1 only $a \parallel b$ holds, whereas for PN2 $a \parallel b$, $a \parallel c$, $b \parallel c$ are true. What is then specific about the latter three instances? They are the only relational knowledge about the PN2 process expressed in terms of $\prec$ and $\parallel$ relations.

In the general case, having no idea whether a described process is distributive or not, one can not reconstruct uniquely the cumulative diagram behaviour from such knowledge because the same three conditions would be true for another, distributive, process such as shown in Fig.4.



PN3

Fig.4

From this problem we can see that the Reisig's "partial ordering" (we rather prefer it to be refered as, say, the "second-order partial ordering") seems, nevertheless, quite reasonable an attempt to deal with non-distributive concurrency.

There is, however, another way to distinguish between PN2 and PN3. This technique, though being rather raw yet, is based on using the so-called "generalizable" $\parallel$-relation.

We define R as a generalized relation[1)] on a set of objects $A = \{a1, a2, \ldots, an\}$. Let R be defined for the set

1) Perhaps, the term "generalized relation" is rather naive for this category.

of subsets of A, rather than just for pairs in A, as usually defined any binary relation, or just for n-tuples, as in the case of an ordinary n-ary relation.

Thus, for our purposes, for example, R(ai, aj, ak) will mean that a certain quality R is true (or takes place) for three objects ai, aj, ak $\in$ A. More specifically, if R = $\|$, then by $\|(a, b, c)$ we may denote that a, b and c are concurrent "in three".

Using this concept of a generalized relation we can observe the following.

Observation 2. For distributive concurrency the $\|$-relation is universally generalizable, i.e.
if $A' \subseteq A$ and $\|$ is a generalized relation,
then $\|(A')$ if and only if $\forall G \subseteq A'$: $\|(G)$.

For our example, for distributive process PN3 we can state that

$$\|(a, b, c) = \|(a, b) \wedge \|(b, c) \wedge \|(a, c)$$

whereas for non-distributive PN2 it is not the case.

◇ The above short analysis points to some notions in relational algebra, particularly, the notion of generalizable relations, which have so far not been seen elswhere by the authors, who would be most pleased to receive any comments or pertinent references on the issue, as well as any new observations on "distributive vs non-distributive" concurrency paradigm.

References

/1/ L. Castellano, et al., Concurrency vs Interleaving: an Instructive Example, EATCS Bulletin 31 (1987).

/2/ D. B. Benson, Concurrency and Interleaving are Equally Fundamental, EATCS Bulletin 33 (Oct 1987).

/3/ W. Reisig, Concurrency is More Fundamental than Interleaving, EATCS Bulletin 35 (June 1988).

/4/ D. E. Muller, Lecture Notes on Asynchronous Circuit Theory, Spring 1961, University of Illinois, Digital Computer Laboratory.

/5/ V. I. Varshavsky, et al., Self-Timed Control of Concurrent Processes,to be published by Kluwer, 1989.