

Shapes of concurrency

*Piotr Chrzastowski-Wachtel
Institute of Informatics
Warsaw University
pch@mimuw.edu.pl*

The question, how true concurrency differs from interleaving, has been studied intensively, beginning with the works of Mazurkiewicz [Maz], Milner [Mil], Pratt [Pra] and many others. The Mazurkiewicz trace theory requests for specifying the dependency relation, which makes such difference explicit. If in some state the sequences ab and ba are executable, then if a and b are independent, then both the sequences describe the same behavior. Moreover, we can say about a concurrent step allowing the execution of a and b in parallel. However, if a and b are dependent, then such two sequences describe two different sequential processes.

Pratt in [Pra] makes an interesting geometrical interpretation of concurrency, letting it be considered as the creation of a new dimension. If we imagine the progress of executing a and b as segments: a on OX axis and b on OY axis, both originating in O, then executing ab in a sequential way means that we first go along the segment a , next come to the corner of a rectangle, from which we traverse the segment b . On the other hand, when a and b are concurrent, we can imagine the space of possible states as the whole interior of the rectangle spanned on a and b as sides.

In many situations, when we consider the reachability graph of a concurrent system, whenever two actions can be executed in parallel, we can see a diamond-like shape.

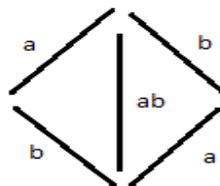


Fig 1. A diamond-like concurrency

The diagonal ab reflects the parallelism. Actions a and b can occur in any order (left-hand and right-hand paths) or they can occur in parallel.

The situation gets more complex if we allow multiple actions to be performed in parallel. The diamond-like pattern can no longer be sufficient. The question of interleaving becomes inappropriate. Let's illustrate this phenomenon on one physical example. From now on we assume that we have ideally resilient identical balls of negligible radius. We will consider collisions without energy loss, satisfying the laws of physics, so exchanging the momentum and energy (perfectly elastic collisions).



Fig 2. Three colliding balls

Let's start with a simple example of three balls on a straight line. The left-hand ball is moving with constant velocity v to the right, the right-hand ball is moving with velocity v to the left, while the middle ball does not move. Let's encode the state of the system by three letters: R for right, S for stand and L for left. So the initial state of the system is RSL.

There are three actions possible in the initial state:

1. The first ball hits the middle one, stopping immediately and transmitting all the momentum to the second ball leading to the state SRL.
2. The third ball hits the middle one, leading to the state RLS

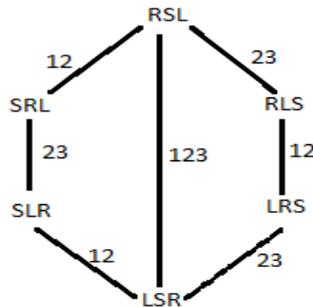


Fig 3. State graph of the three colliding balls

3. The first and the third ball hit the middle one, bouncing backwards, without moving it, so leading to the final state LSR. Two first of them lead to

consecutive events that must happen next. The complete picture of the states is given on Fig 3. The labels on edges name the balls, which take part in the collision. Mind that they reflect different actions (for instance 12 means just that the ball 1 collides with ball 2, without specifying what kind of collision it is).

The diagonal labeled 123 reflects the event in which the first and the last ball hit the middle one in the same moment. In fact it can be interpreted as $12 \parallel 23$, but this time no interleaving is possible. The event 123 is not a combination of 12 and 23.

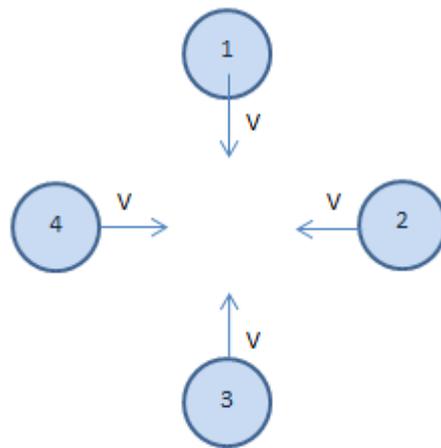


Fig 4. Four colliding balls

Such hexagonal shape can be typical for 3 objects, which interact in more complex situations, like one billiard ball hitting the two other ones, which stick together; a situation of great instability.

We can presume that the multi-event parallelism can create much more complex patterns. Let's consider one more example. Imagine 4 balls, like on Fig 4, targeting to one common centre: ball 1 going South, ball 2 West, ball 3 North and ball 4 East. We do not assume that the balls are equidistant from the center. It can happen that, for instance, ball 1 hits ball 2 before any other event happens. According to the laws of physics, in this case ball 1 and ball 2 will exchange the momentum, hence ball 1 will start going West, and ball 2 will go South. Observe, that if any collision happens, it will take place in the common center and the resulting directions will be like the original ones, so S-W-N-E. Eventually we expect that the four balls, after possible bounces, will pass the center and leave the initial area going to infinity.

Let us encode the states by four letters assigned to each of the balls, so the initial state will be SWNE.

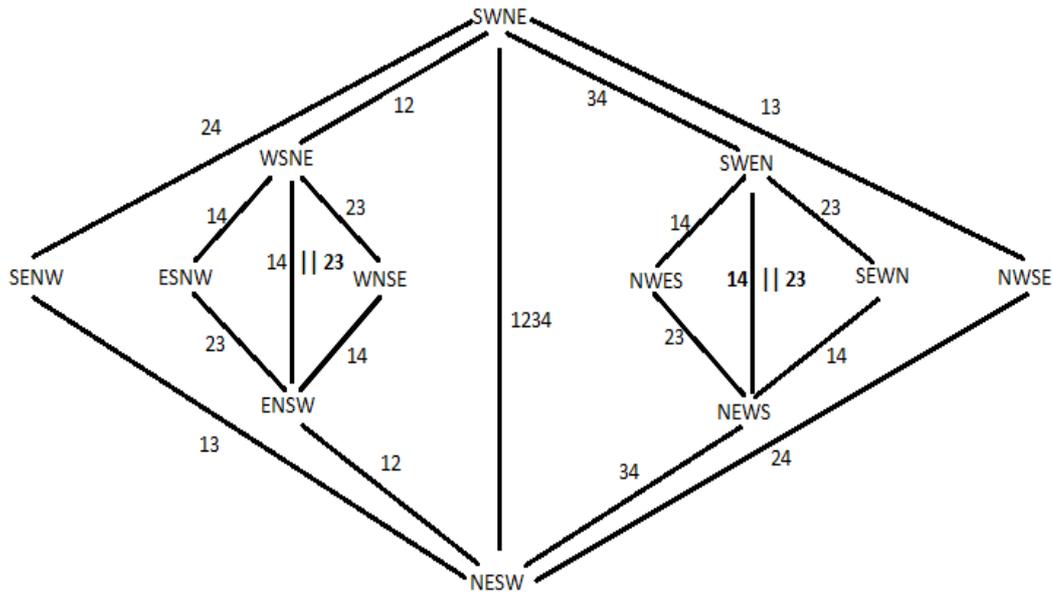


Fig 5. Part of the state graph of four possibly colliding balls

On Fig. 5 we can see a part of the big state graph, again hiding some information, like where the ball actually is with respect to the center. Some of the possible runs are not shown, like the one, in which none or only one collision takes place.

In the last case we can see an octagonal shape (among many) SWNE-WSNE-WNSE-ENSW-NESW-NEWS-NWES-SWEN, which makes sequentially the same result as the main diagonal SWNE-NESW. The diagonal run reflects the situation, in which all the balls are equidistant and reach the center at the same time. Each of them will bounce symmetrically the two neighboring ones and go immediately, without intermediate steps into the final direction. Mind that inside the diagram we can see two typical diamond-like occurrences of concurrent events.

So again, here we cannot replace the concurrent behavior by the composition of the sequential ones. Concurrency, like in the previous example, creates new run, which has little to do with the partly sequential ones, leading to the same result.

On Fig 5 we also see an interesting case of a diamond non-concurrency: the outer edges, however they form a diamond shape, cannot be replaced by a diagonal 13 || 24. Bouncing ball 1 against ball 3 can never happen in the same moment as bouncing ball 2 against ball 4.

With the increase of the number of parallel events, one can expect other shapes of concurrency, not necessarily being planar. Usually they reflect unstable physical systems -- a small change in the initial state can cause a dramatic change in the behavior. Such instability is well recognized by the physicists. Let us look deeper into some other part of the state graph of the behavior of the system from Fig. 4. For the moment let us concentrate on three balls 1,4,2, assuming that the ball 3 is far away from the common center. So the initial state of these balls is SWNE, but the ball 3 will not change its status for a while.

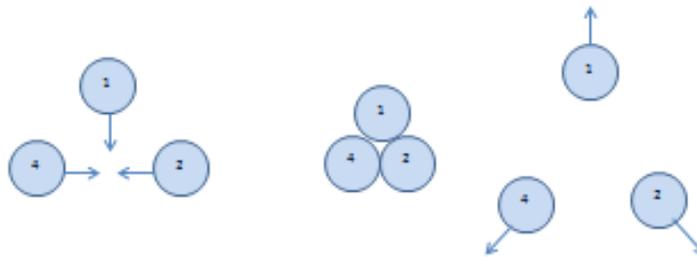


Fig.6 Unstable collision of 3 balls

If all the three considered balls collide in the center simultaneously, then the ball 1 will go North, but the two balls 4 and 2 will exchange their momentums and absorb the momentum of ball 1. As a result they will bounce in two directions a bit South, as depicted in Fig.6. Let us call this state N(SE)N(SW) No other collision would happen, so this would be a final state of the system.

This will differ a lot from the case, in which the ball 2 would find itself a little bit towards East in the initial position. In such a case two collisions between the balls 1,4,2 would happen. First the ball 1 would collide with the ball 4 exchanging their momentums and transforming the system state to EWNS, next the ball 1 and 2 would collide reaching the state WENS, quite different from the final state of Fig 6. So a small change in the initial position would result in a major qualitative difference of the final state. This is an illustration of an instability phenomenon, in which case the multi-collision of balls cannot be factorized as a result of interleaving the bi-collisions.

Anyway, the world of concurrency seems to hide some mysteries. Here we demonstrated a set of examples showing why interleaving models can be not appropriate for modeling true concurrency.

Bibliography

[Maz] A.Mazurkiewicz, Introduction to Trace Theory, in The Book of Traces; V.Diekert, G.Rozenberg (Eds.) World Scientific, 1995

[Mil] R. Milner, *Communication and concurrency*, Prentice Hall International Series in Computer Science, 1989

[Pra] V.R.Pratt, *Modeling Concurrency with Geometry*, in 18th Annual ACM Symposium on Principles of Programming Languages, Orlando, Florida, USA, 1991