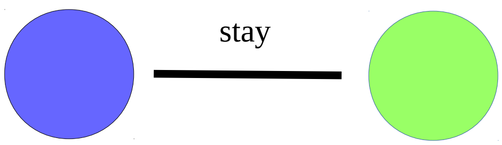
Billiard Balls in Asymmetric Velocity Logic

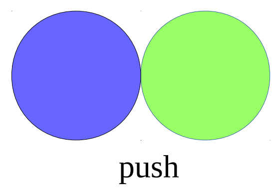
by Sven Nilsen, 2020

In this paper I explain asymmetric velocity logic using examples with billiard balls.

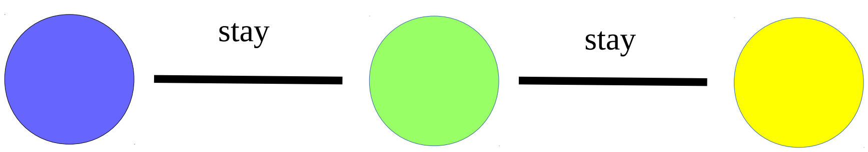
Asymmetric velocity logic is a theory that formalizes a lot of common sense problems in physics. Instead of describing the full state of physical objects, one uses asymmetric velocity logic at a higher level that concerns the dynamic velocity constraints between objects.

Alice has two billiard balls. When the balls are not touching each other, they have the velocity constraint “stay”. This means that if one of the balls are moving, it looks like the other ball is moving seen from the ball’s velocity reference frame.





However, when the balls are touching each other, the moving ball can not go closer to the other ball, but it can move away from the other ball. This is represented as the velocity constraint “push”.



Alice places three billiard balls with space between them on a line. She wants to prove that the blue ball and the yellow ball have a velocity constraint “stay” between them.

The “stay” velocity constraint has an associated function `bool → bool`:

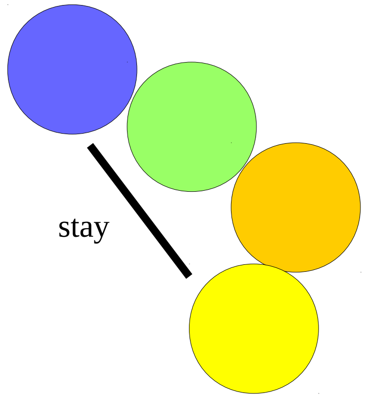
stay <=> \false

Logical AND (the `∧` symbol) with Higher Order Operator Overloading is used to join paths:

∵	stay ∧ stay	joining paths from (blue, green) to (green, yellow)
∴	\false ∧ \false	
∴	\false	
∴	stay	

This proof technique can be used when the billiard balls are ordered in one dimension.

In two dimensions, there can be multiple paths from one billiard ball to another.



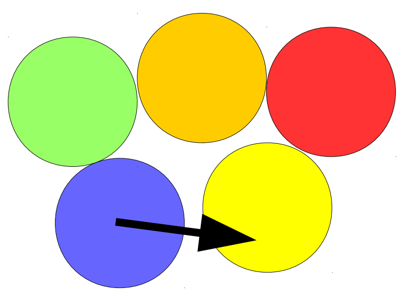
∵ (blue, green) = push
∵ (green, orange) = push
∵ (orange, yellow) = push
∵ (blue, yellow) = stay

Logical OR (the \`∨\` symbol) with Higher Order Operator Overloading is used to meet paths:

∴ (push ∧ push ∧ push) ∨ stay <=> push

Therefore, if the blue ball moves toward the yellow ball, it will push the yellow ball.
This is true even the blue ball and the yellow ball do not directly touch each other.

Counter-intuitively, this logic even works in the follow scenario:



In this example, the blue ball can move closer to the yellow ball without pushing it.

However, by doing so, the blue ball will no longer touch the green ball:

∵ (blue, green) = **stay** The green ball will no longer be pushed
∵ (green, orange) = push
∵ (orange, red) = push
∵ (red, yellow) = push
∵ (blue, yellow) = stay

∴ (**stay** ∧ push ∧ push ∧ push) ∨ stay <=> stay