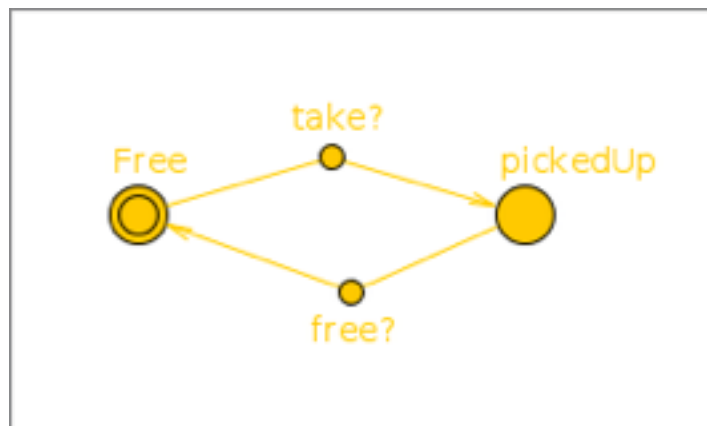


This is the philosopher model, it starts in a resting state and can change between thinking and resting states freely. When a philosopher wants to eat he switches to wantsToEat, this increments a counter to let the system know the philosopher is in a hungry state. The next option for him to do is check if one of his adjacent forks is

available, it also checks if taking this fork will place the system in deadlock or if we are depriving another philosopher and causing him to starve, giving him priority to go ahead of us. We also increment the state of the philosopher to know that we added 1 fork to him. The process is repeated in the next state while picking up our second fork, where we can stay in the eating state for a while. When the philosopher is done eating we lay down both forks, and reset the philosophers state to 0 where he is either resting or thinking and will go back to 1 when he decides he wants to eat again.



This is fork model, we simply take the fork if the checks in the philosopher model pass and is moved to a pickedUp state and then frees when needed.



When initializing the system the parameters are important, representing their position around the table.

The positions are shown in the image to the left, and if more philosophers were added to the table (as well as fork per respective philosopher) this model should support it if positions are provided following this pattern.