The amoebot model is a distributed computing model of programmable matter. It envisions programmable matter as a collection of computational units called ``amoebots'' that utilize local interactions to achieve tasks of coordination, movement, and conformation.
The majority of existing works using the amoebot model have treated concurrency at a high level.
Under the simplifying assumptions that all amoebot actions are atomic, isolated, and reliable, existing algorithms have been analyzed as if they are executed in a fair serial setting where only one amoebot is active at a time.
In my presentation, I will refine the amoebot model to directly treat the concurrent setting, defining a new execution model based on granular operations that compose amoebot actions.
I will then identify sufficient conditions for guaranteeing the correctness of serial algorithms in the concurrent setting and present a concurrency control protocol which, given any algorithm that is correct in the serial setting and satisfies some basic conventions, produces an equivalent algorithm that satisfies the sufficient conditions for correctness in the concurrent setting.
It is, of course, possible to define algorithms that are correct in the concurrent setting without our protocol and its conventions, as I will demonstrate with a new formulation of the hexagon formation algorithm (NANOCOM '15), but this adds significant complexity to algorithm design and analysis.