

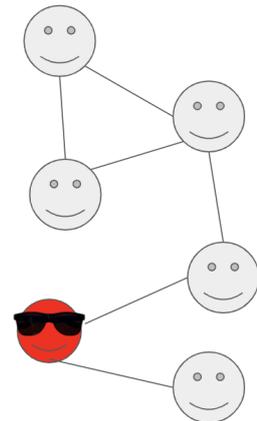


Topics in Approximate Agreement

Distributed Computing focuses on using several processors to solve a problem cooperatively. Considering that Murphy's Law is well known to apply to reality, achieving meaningful results in distributed algorithms even in the presence of faults becomes essential. Such faults may be permanent, and it may be that processes deviate arbitrarily from the protocol and are controlled by a central adversary (*Byzantine* faults).

One of the essential building blocks in Distributed Computing is *Byzantine Agreement*, where each processor holds some input value, and they must output the same value such that a specific validity condition is satisfied (for example, this output must be the input of one honest processor).

It is well known that Byzantine Agreement can only be achieved in synchronous networks, where messages are guaranteed to be delivered within a known amount of time. The seminal result of Fischer, Lynch, and Paterson showed that in the asynchronous setting, where messages may be delayed arbitrarily long, achieving Byzantine Agreement is impossible even when only one processor may crash. This negative result led to multiple relaxations of the original definition of Byzantine Agreement that can be achieved in the asynchronous setting. A variation we focus on is *Approximate Agreement*: a problem based on multi-valued Byzantine Agreement that allows the processors to output ϵ -close values instead of requiring them to output the same value.



There are quite a few problems left to be solved in Approximate Agreement: especially when it comes to optimal runtime, or extending the setting from real-valued inputs to other input spaces.

Requirements: Motivation and interest in theory are required. Knowledge in cryptographic protocols is a huge plus. Being able to work independently and coming up with your own ideas are highly appreciated.

Interested? Please contact us for more details!

Contact

- Diana Ghinea: ghinead@ethz.ch, ETZ G97