1. **Clocks and Oscillators in Population Protocols  
   [Specification and Proposed Design]**

Sanchita Roy

Student Id - 201594919

Primary Supervisor - Leszek Gasieniec

1. **Project Description:**

Population protocol models are a fairly recent development. Fundamentally, a population of miniature mobile agents that interact with one another to do a computation is characterized as population protocols. The agents are finite state machines with identical programming. The agents are first given input values, and pairs of agents can communicate when agents are close to one another, they should exchange information. Although the agents' movement is random, there is some justice limitations, calculations, and the right result must ultimately converge any timetable that emerges from that movement will be valuable.

The population protocol model was developed to depict sensor networks made up of a small number of immobile entities that are unable to control their own motion. It is also strikingly similar to theoretical chemistry models of interacting molecules.

Computability: When researching which functions can be computed by population protocols, one can emphasis on computing predicates.

One-way communication: Angluin et al. investigated a number of weaker interaction models in which information only travels in one direction during interactions. The status of a sender agent is revealed to a receiver agent, but the sender is unaware of the receiver's condition. A system with such one-way communication is only as powerful as the specifics of the communication mechanism.

Interaction Graphs: Physical restrictions on agents' mobility may exist in particular circumstances, which will restrict the interactions that may take place. This data is represented by an interaction graph, where nodes are agents and edges are potential interactions.

Random Interactions:   The most basic version of this type assumes uniform random interactions, where each pair of agents has an equal chance of interacting at each phase. The first population protocol paper by Angluin et al. included protocols for random scheduling.

Failures: Some assumptions could be unrealistic in the context of mobile systems of small agents; an effort was made to start with a clear model. Some work has studied fault tolerant population protocol like, Crash failures, Byzantine failures.

1. **Aims and Objectives:**

The main aim in this project is to study, implement and experiment with periodic mechanisms in population protocols including phase clocks and oscillators. Such mechanisms allow to simulate simple biological processes as well as to synchronize more complex distributed computing processes.

In this project we will focus on basic oscillating processes including:

* Cycling progress based on Rock-Paper-Scissor game
* Leader based phase clocks
* Leaderless phase clocks also known as oscillators

We will also consider sub-variants of the above listed periodic processes which will reflect the needs of specific biological processes, e.g., pray vs predator, hormone regulation mechanisms and others.

1. **Key Literature and Background Reading:**

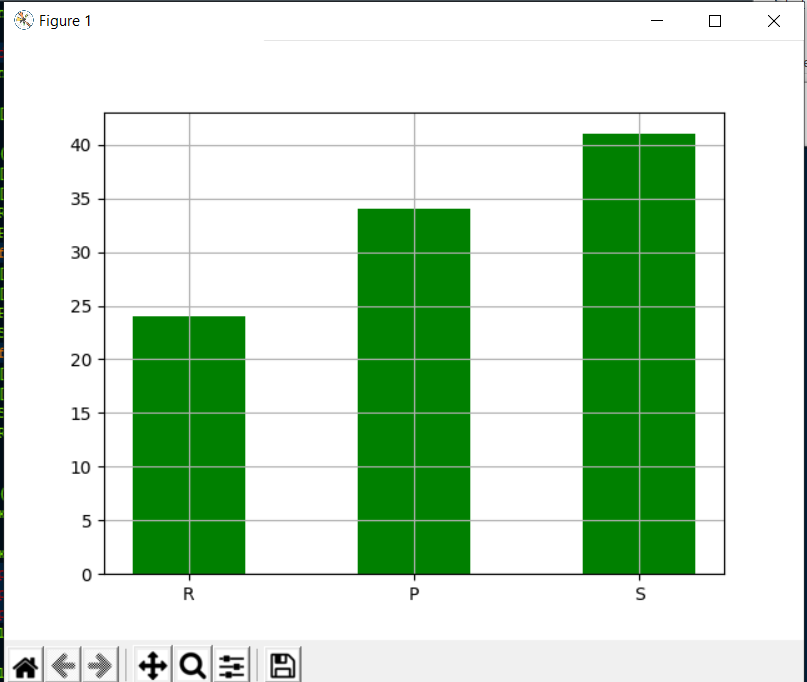
Below mentioned are some of the documents I have studied for background reading.

* Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, René Peralta, Computation in networks of passively mobile finite-state sensors, Distributed Computing 18(4): 235-253 (2006).
* James Aspnes, Eric Ruppert: An Introduction to Population Protocols, Middleware for Network Eccentric and Mobile Applications, Springer, 2009, pp 97–120.
* Colin Cooper, Anissa Lamani, Giovanni Viglietta, Masafumi Yamashita, Yukiko Yamauchi, Constructing self-stabilizing oscillators in population protocols, Information and Computation (255/3), August 2017, Pages 336-351.
* Jurek Czyzowicz, Leszek Gąsieniec, Adrian Kosowski, Evangelos Kranakis, Paul G. Spirakis, Przemysław Uznański, On convergence and threshold properties of discrete Lotka-Volterra population protocols, JCSS (130), December 2022, Pages 1-25.
* Adrian Kosowski, Przemysław Uznański, Population Protocols Are Fast, PODC 2018: 475-477.
* ADD HERE SOME POSITIONS ABOUT PROGRAMMING IN PYTHON AND STUDENT PROJECTS

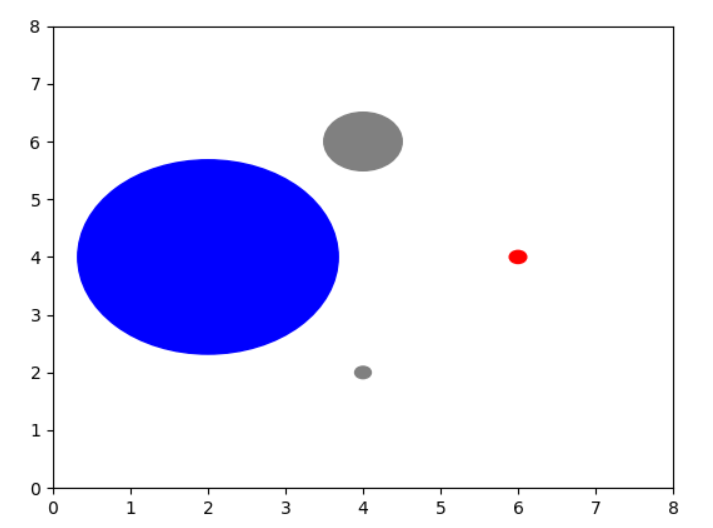
1. **Development and implementation Summary:**

The project involves several parts including initial studies on population protocols, choice a programing language, design of the solution, implementation, experimentation and the final analysis.

1. Initial studies will involve good understanding of population protocols and what has been done in the context of phase clocks and oscillators
2. We decided to use Python to run the experiments, mainly due to easy access to libraries supporting math operations and visualization
3. The proposed design is relatively simple, and it comprises implementation of population protocol environment (already done), visualization (partially done), and computation of the relevant statistics (in development).
   1. Agents in population are represented as simple objects. Pairs of such objects are chosen uniformly at random by the tools available in python. The rules governing periodic processes are implemented as cases (complementing if statements). The inputs are predefined in the initial configuration, e.g., in Rock-Paper-Scissor game in the input three relevant populations of R, P and S are of the same size, where the rules are (R,P) -> (P,P), (P,S) -> (S,S), and (S,R) -> (R,R). During the execution of this game the sizes of R, P, S change chaotically to eventually stabilize on one of the configuration (R,0,0), (0,P,0) or (0,0,S). In other considered problems we will study more permanent oscillating processes where a stable configuration is never reached.
   2. Visualization is done with the help of matplotlib library available in Python. You can see the first attempt in Figure 1. More adequate visualization based on ball-like representation of “hours” of the clock are shown in Figure 2. The final format of the visualization will be delivered on the conclusion of experimental work.
   3. The relevant statistics will be gathered for all considered processes have more formal understanding of the mechanics of oscillation.
4. The final analysis and conclusion will be done to summarize the work performed within this project and potential further work to be done by others.



A system based on two species feeding on one another



1. **User Interface Mockup:**
2. **Data Sources (***not known as such***)**

We will generate the resources artificially by manipulation of initial configurations in the considered processes.

1. **Testing:** *(Not sure, how it’ll be implemented in this project)*

The testing will be mostly focused on two aspects: correctness and good (meaningful) visual presentation. The correctness will be verified via multiple runs of typical cases and closer studies of very small and border cases. The visual aspects will be assessed by my own and other user perception, and its alignment with the collected statistical data.

1. **Evaluation:**

The evaluation will mainly focus on correctness, visual presentation, the learning aspect and the success of the project.

1. **Ethical Consideration:**

All input data (configurations) will be synthetic. No special ethical considerations beyond own work and full professionalism are to be considered.

1. **Project Plan:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TASK NAME** | **START DATE** | **END DATE** | **DURATION\* (WORK DAYS)** | **PERCENT COMPLETE** |
|  |  |  |  |  |
|  |  |  |  |  |
| Background reading and Literature review | 6/6 | 6/17 | 12 | 100% |
| Specification and Design report Submission | 6/20 | 7/22 | 33 | 60% |
| Implementation | 7/25 | 8/19 | 26 | 20% |
| Final Presentation slides | 8/22 | 9/2 | 12 | 10% |
| Dissertation | 9/5 | 10/7 | 33 | 0% |

1. **Risks and Contingency Plans:**

The main emphasis is on experimentation with several periodic processes. If I do not have enough time the parts related to visualization and the interface will be reduced in order to at least get meaningful statistics related to experiments.

1. **References:**

Adrian Kosowski, Przemysław Uzna´nski: Population Protocols Are Fast

ColinCoopera, AnissaLamanib, GiovanniVigliettac, MasafumiYamashitab, YukikoYamauchib: Constructing self-stabilizing oscillators in population protocols

James Aspnes, Eric Ruppert: An Introduction to Population Protocols