HAL
open science

Population protocols, games, and large populations

Xavier Koegler

To cite this version:

Xavier Koegler. Population protocols, games, and large populations. Networking and Internet Architecture [cs.NI]. Paris Diderot Unviersity, 2012. English. NNT : . tel-01274140

HAL Id: tel-01274140

https://inria.hal.science/tel-01274140v1

Submitted on 15 Feb 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Université Paris Diderot – Paris 7
Laboratoire d'Informatique Algorithmique: Fondements et Applications

Population protocols, games, and large populations.

Thèse présentée pour l'obtention du diplôme de

Docteur de l'Université Paris Diderot, spécialité Informatique

à l'École Doctorale de Sciences Mathématiques de Paris Centre

Par

Xavier Koegler

Thèse dirigée par Olivier Bournez et Pierre Fraigniaud

Soutenue publiquement
le 13 septembre 2012
devant le jury constitué de :

Directeurs de Thèse :	M. Olivier Bournez	Professeur
	M. Pierre Fraigniaud	Directeur de Recherche
Raporteurs :	M. James Aspnes	Professeur
	M. Bruno Gaujal	Directeur de Recherche
Examinateurs :	M. Vincent Blondel	Professeur
	M. Pascal Koiran	Professeur
	M. Jean Mairesse	Directeur de Recherche

Résumé :
Le modèle des *population protocols* a été proposé pour capturer les spécificités de réseaux opportunistes constitués d'une population d'agents mobiles à la mémoire limitée capables de communications sans fil par paires. L'objet de cette thèse est d'étendre la compréhension et l'analyse des *population protocols* ainsi que leur liens avec d'autres modèles de dynamiques de populations.

La première contribution de cette thèse est l'étude de la traduction en terme de protocoles de population de la dynamique d'une population d'agents jouant à un jeu de manière répétée les uns contre les autres et adaptant leur stratégie selon le comportement de *PAVLOV*. Nous montrons que les protocoles issus de tels jeux sont aussi puissants que les protocoles de population généraux.

La deuxième contribution consiste à étudier des hypothèse de symétrie dans les jeux et dans les transitions d'un protocole de population, pour montrer que, si les protocoles de population symétriques sont équivalents aux protocoles généraux, les jeux symétriques sont, eux, significativement moins puissants.

La troisième contribution est de montrer comment étudier le comportement d'une protocole de population lorsque la taille de la population tend vers l'infini en approchant la dynamique résultante à l'aide d'une équation différentielle ordinaire et de définir un calcul par grande population comme la convergence de cette équation différentielle vers un équilibre stable.

La quatrième et dernière contribution de la thèse est la caractérisation des nombres calculables en ce sens comme étant très exactement les réels algébriques des $[0, 1]$.

Abstract:
Population protocols were introduced to capture the specifics of opportunistic networks of tny mobile agents with limited memory and capable of wireless communication in pairs. This thesis aims at extending the understanding and analysis of *population protocols* as well as their links to other models of population dynamics including ones from game theory.

The first contribution of this thesis is to translate in terms of population protocols the dynamics of a population of agents playing a game repeatedly against each-other and adapting their strategy according to the *PAVLOV* behaviour. We show that protocols born from games are exactly as powerful as general population protocols.

The second contribution consists in the study of the impact of symmetry on games and in the transitions of a population protocol to show that, if symmetric population protocols are equivalent to general protocols, symmetric games are significantly less powerful.

The third contribution is to show how the dynamic of a population protocol can be approximated by an ordinary differential equation when the population grows to infinity. We then define a computation by a large population to be the convergence of this differential equation to a stable equilibrium.

The fourth and final contribution of this thesis is the characterisation of the numbers computable in the above sense as exactly the algebraic real numbers in $[0, 1]$.

Remerciements

Je tiens tout d'abord à remercier mes deux directeurs de thèse, Olivier et Pierre, sans lesquels rien de ce travail n'aurait été possible. Si leur encadrement et leur suivi ont été deux appuis essentiels aux cours de cette thèse, je garderai toujours un souvenir exceptionnel des réunions de travail à trois, à échanger des idées et des pistes, échafauder des démonstrations au travaux et nous contredire les uns les autres.

Olivier m'a accompagné depuis mes premiers pas dans la recherche, il y a 6 ans déjà, en stage de licence avec Johanne Cohen, ils m'ont donné le goût de la recherche, de l'algorithmique distribuée et des protocoles de population qui m'ont amené à faire cette thèse. Je sais que je ne suis pas le seul stagiaire passé entre leurs mains à avoir poursuivi en thèse dans le domaine, voire même avec eux et c'est là un gage certain de leurs qualités tant humaines que scientifiques. Depuis le Master 2, Pierre s'est joint à la petite équipe et je ne peux que m'en féliciter. Sa gentillesse et sa bonne humeur m'ont autant aidé pendant ces années que ses connaissances et son intelligence.

Tous deux ont toujours su trouver le temps, malgré la lourde charge qu'est la direction d'un laboratoire, et les exigences de leurs autres étudiants et stagiaires, de m'écouter, de m'aider et de me guider. Leur compagnie a également été l'occasion de très agréables soirées lors des conférences et réunion de projets et je n'aurais pu souhaiter de meilleurs encadrants.

Je voudrais exprimer ma reconnaissance James Aspnes et Bruno Gaujal d'avoir bien voulu relire ce manuscrit, m'en signaler les erreurs et imprécisions. Leurs commentaires m'ont permis de grandement améliorer ce document. Je suis également reconnaissant Vincent Blondel, Pascal Koiran et Jean Mairesse d'avoir bien voulu participer au jury de ma thèse.

Je remercie aussi tous ceux, au LIAFA dont le travail m'a permis d'effectuer cette thèse dans de bonnes conditions. Tout particulièrement Noelle et Nathalie pour leur patience et leur aide pour les taches administratives ainsi que Houy et Laifa pour la partie informatique. Je remercie aussi l'ensemble du LIAFA pour la bonne ambiance dans le laboratoire. Je remercie particulièrement Amos Korman et Ofer Feinerman pour l'opportunité qu'ils m'ont offerte de travailler avec eux.

Je suis tout particulièrement reconnaissant à ceux qui ont partagé mon bureau et en ont fait un lieu chaleureux et convivial. Denis, sans lequel nous n'aurions sans doute jamais eu de canapé, Thach pour les concerts, Hervé qui était là déjà dans le cagibit du stage de Master, Charles et Heger. Sans oublier ceux qui sont partis avant, Anh et sa guitare, Mauricio, George, Vincent ou Juraj.

Je remercie les membres des projets ANR ALADDIN, DISPLEXITY et PROSE qui,

en réunion, ont pu me faire découvrir d'autres champs scientifiques ou me suggérer d'autres pistes pour mes travaux, ou tout simplement me faire passer un bon moment. En particulier, Fabien Mathieu et ses gadgets toujours renouvelés et Émilie Diot.

Je tiens également à remercier ceux qui, en dehors de la sphère professionnelle, m'ont accompagné ces trois années. Le Krou tout d'abord, Damien, Léo, Florian, Marion et Simone, pour trois années passées à vivre, à regarder les JO d'hiver en pleine nuit, à discuter d'informatique, de sciences, de tout et à ce soutenir les uns les autres. Monkeytown et associés, trop nombreux pour être nommés individuellement, Alois, Guilhem, Laura, Maxime, Sami et les autres.

Je remercie bien entendu mes Parents, qui m'ont très tôt donné le goût des sciences et ma famille qui a su prêter une oreille patiente à mes explications, parfois difficilement compréhensibles, sur mes recherches.

Enfin, je remercie Catherine de son soutien et de son affection, surtout pendant les derniers mois, les plus difficiles, de l'élaboration de cette thèse.

Acknowledgements.

I would first and foremost like to thank my advisors, Olivier and Pierre without who non of this would have been possible. If their direction and support have been key foundations for this work, I will always keep a fond memory of working with them, exchanging ideas and leads, building and rebuilding proofs, and contradicting one another.

Olivier has been helping me since I first started doing research in Computer Science, 6 years ago during a summer internship he co-directed with Johanne Cohen. Together they fostered my taste for research, distributed systems and population protocols which directly lead to this thesis. I know I am not the only intern who, after passing through their hands decided to continue working in the field, or even with them, for a PhD and I believe this is the strongest illustration of their human and scientific qualities. Since my Master thesis, Pierre has joined us and I can only be grateful. His kindness and good disposition were as helpfull to me as his scientific insighit.

Despite the heavy load of running their respective labs and directing other students, Pierre and Olivier both always found the time to help me, listen to me and guide me. Their company was also very pleasant during those evenings, at conferences or project meetings and I could not have wished for better advisers.

I would like to thank James Aspnes and Bruno Gaujal who were kind enough to review this manuscript and point out the mistakes, imprecisions and typos therein. Their comments allowed me to greatly improve this document. I am also grateful to Vincent Blondel, Pascal Koiran and Jean Mairesse who accepted to sit on the jury for my defence.

I would also like to acknowledge all those, at LIAFA, whose work allowed me to pursue my research in the best conditions. Especially Noelle and Nathalie for the administrative side and Houy and Laifa for IT. I thank the whole of LIAFA for the cordial atmosphere in the lab. I am quite grateful to those who shared my office and made it a pleasant and warm working place. Denis, without whom we probably would not have a couch, Thach for the concerts, Hervé who was there in the early cupboard office, Charles, and Heger, as well as those who have left already: Anh and here guitar, Mauricio, George, Vincent or Juraj. The were all part of a wonderful environment.

I would also like to acknowledge the members of ANR projects ALADDIN, DISPLEXITY and PROSE who showed me other research topics and suggested different approaches to my work, or simply were good company. Notably Fabien Mathieu and his ever renewed gadgets and Émilie Diot.

I would also like to thank those who, outside the worksphere were with me for these

last three years. The Krou, first: Damien, Léo, Floran, Marion and Simone who where wonderful company for three years living, watching the Winter Olympics in the middle of the night, talking computers, science and generally supporting each other. Monkeytown and its associate, too many to enumerate, Alois, Guilhem, Laura, Maxime, Sami and all the others. Thank you my friends.

I would of course like to thank my Parents, who gave me the science bug from a very young age and my family for being patient and understanding when I tried, sometimes not in an easily understood way to explain my research.

I would finally like to thank Catherine for her love and support, especially in the last and hardest few months of this thesis.

Contents

Introduction [FR] **9**
Contexte et Motivations . . . 9
Contributions . . . 11

Introduction [EN] **15**
Context and Motivations . . . 15
Contributions . . . 17

1 Population Protocols **19**
1.1 Population protocols and their computational power . . . 20
1.1.1 Definitions . . . 20
1.1.2 Computational power. . . . 21
1.2 Random Scheduler . . . 22
1.3 Restricted Interactions . . . 23
1.3.1 One-way interaction . . . 23
1.3.2 Restricted communication graphs . . . 24
1.4 Enhanced individual computational power . . . 25
1.4.1 Unique identifiers . . . 25
1.4.2 Passively mobile machines . . . 25
1.5 Fault tolerance . . . 25
1.5.1 Byzantine Agents . . . 26
1.5.2 Crashes and Transient Failures . . . 26
1.6 Self Stabilization . . . 27

I Games and Population Protocols **29**

2 Pavlovian Population Protocols **33**
2.1 Elementary Game Theory . . . 33
2.2 From Games To Population Protocols . . . 35
2.3 Main Result . . . 37
2.4 Threshold Predicates . . . 39
2.5 Modulo Counting . . . 42

2.6 Conclusion . . . 46

3 Symmetric Games and protocols **49**
3.1 Symmetric population protocols . . . 50
3.2 Some simple exclusive Pavlovian protocols . . . 56
3.2.1 Counting up to 3 exclusively. . . . 56
3.3 Some non-trivial Exclusive Pavlovian Protocols. . . . 59
3.3.1 Leader Election . . . 59
3.3.2 Majority . . . 60
3.3.3 Counting up to 2^k . . . 61
3.4 Conclusion . . . 66

II Computing with Large Population Protocols **69**

4 Large Population Protocols **73**
4.1 An illustrative example . . . 74
4.1.1 A General Theorem about Approximation of Diffusions . . . 76
4.1.2 Proving convergence of our example. . . . 77
4.1.3 Giving a better Asymptotic Development of $p(k)$ . . . 79
4.2 General Framework . . . 80
4.3 Computing with LPPs. . . . 86
4.4 Conclusion . . . 87

5 The computational power of LPPs **89**
5.1 Any computable number is algebraic. . . . 89
5.2 Computing Algebraic Numbers . . . 90
5.2.1 Computing Rationals . . . 91
5.2.2 Derandomization . . . 91
5.2.3 Constructing Equilibria . . . 94
5.2.4 Enforcing Stability . . . 96
5.3 Conclusion . . . 100

Conclusion and Perspectives **103**
Conclusion . . . 103
Perspectives . . . 104

Introduction

Cette thèse traite de modèles de calcul dans des réseaux *opportunistes* constitués de petits agents mobiles, capables de communication sans fil avec des interactions par paires. De tels réseaux sont caractérisés par l'apparition sporadique des connections entre les agents que nous qualifions de *passivement mobiles*. C'est à dire que les agents (1) ne maîtrisent pas leur propre déplacement, mais sont en réalité mus par une force échappant à leur contrôle et au contrôle de leur créateur et (2) sont capable d'interagir avec d'autres agents passant à proximité en échangeant de l'information via un protocole de communication non spécifié. Un autre aspect important de ces modèles et que tous les agents doivent être programmés de manière identique. En conséquence, les agents sont entièrement anonymes et indistinguables du point de vue d'un observateur extérieur, la seule différence étant dans la quantité restreinte d'information conservée dans leur mémoire individuelle, ce qui inclut, pour chaque agent, une valeur d'entrée provenant d'un ensemble fini d'entrées possibles.

Plus précisément, nous considérons le modèle des *population protocols* (ou *protocoles de population*) introduit par Angluin, Aspnes, Diamadi, Fischer et Peralta dans [2] afin de décrire les spécificités de certains réseaux de capteurs. Ce modèle a été sélectionné pour ses hypothèses minimalistes sur la puissance de calcul individuelle des agents ainsi que l'absence totale de contrôle sur le réseau de communication. Les agents d'un protocole de population sont des automates finis capables de communication sans fil qui n'ont ni identifiants propres, ni contrôle sur leur mouvement et sont programmés uniformément en une population de taille finie mais inconnue. Ils communiquent avec les autres agents par paires, sans utiliser de mécanisme de mémoire partagée, simplement en s'informant l'un l'autre de leur état actuel et en mettant leurs états à jour en selon une règle prédéterminée qui constitue le programme du protocole. Deux agents étant dans le même état interne sont entièrement indistinguables par un observateur externe. Les calculs sont effectués par ces protocoles en assurant que l'ensemble de la population converge vers une configuration dans laquelle une propriété désirée est préservée. Par exemple, on pourrait souhaiter que tous les agents se mettent d'accord sur une valeur de sortie, ou qu'une proportion prédéterminée d'agents soit dans un état particulier.

Contexte et Motivations

Cette thèse s'inscrit dans le contexte général des modèles théoriques capturant des dynamiques de populations. De tels modèles ont été introduits dans divers champs scientifiques,

allant de la biologique à la théorie des jeux ou à l'économie. Un même modèle pouvant fréquemment être utilisé dans plusieurs domaines. Les équations de Lotka-Volterra, par exmeples, furent introduites par Alfred J. Lotka dans le contexte des réactions chimiques périodiques [41] puis utiliées plus tard pour modéliser des systèmes proie-prédateur [40]. Une présentation de plusieurs tels modèles issus des sciences naturelles ainsi que de leur analyse mathématique est présentée dans [45].

Les dynamiques de population sont également un sujet d'étude important en théorie des jeux, et tout particulièrement dans le cadre de la théorie des jeux évolutionnaire [49, 33] qui est l'étude de l'évolution d'une population d'agents jouant de manière répétée à un jeu les uns contre les autres et adaptant leur stratégie afin de maximiser une utilité individuelle. De telles dynamiques peuvent fréquemment être utilisées pour capturer des dynamiques de populations issues d'autres domaines. Les équations de Lotka-Volterra, par exemple, sont équivalentes à certaines dynamiques de réplication issues de la théorie des jeux évolutionnaire [34]. Une dynamique de réplication, est une dynamique dans laquelle un agent change de stratégie en fonction de l'écart entre sa réussite personnelle et la réussite moyenne (appelée *fitness*) de la population.

La majeure partie de la littérature sur les dynamiques de population se concentre sur l'étude analytique des différents modèles et systèmes dynamiques, visant à déterminer si une dynamique particulière est ultimement convergente vers un équilibre stable ou présente des orbites stables non ponctuelles ainsi que les propriétés de tels équilibres et de telles orbites. Dans l'esprit de la calculabilité, nous nous concentrerons au contraire sur la possibilité de *programmer* une population d'agents afin d'assurer un comportement particulier de la dynamique résultante. Par exemple, nous pourrons programmer les agents afin d'assurer que l'ensemble de la population converge vers une configuration stable dans laquelle un consensus existe entre tous les agents sur une unique valeur de sortie commune, fonction de la configuration de départ. Une autre possibilité consiste à vouloir forcer à ce qu'une proportion prédéterminée d'agents soit dans un état particulier. Ces travaux peuvent considérés comme une extension du modèle des protocoles de population introduit par Anglin *et al.* dans [2], et nous présenterons un survol des résultats connus au sujet des protocoles de population dans le chapitre 1. Les équations de Lotka-Volterra sont équivalentes à une sous-classe de protocoles de population construite dans le contexte de la théorie évolutionnaire des jeux [21]. Nous nous efforcerons d'étudier de façon plus générale de quelle manière il est possible des jeux permettant de calculer n'importe quel prédicat calculable par un protocole de population à l'aide des dynamiques de *PAVLOV*. Nous considérerons également comment programmer une grande population afin de calculer des nombres réels en forçant une stabilisation de la population sur des configuration où la proportion d'agents dans des états marqués est le nombre calculé.

La tâche consistant à forcer une proportion fixe d'agents dans un état marqué est similaire à la problématique du *slicing* qui consiste à séparer une population en tranches (*slices*) suivant une règle prédéterminée. La plupart des résultats connus sur le *slicing* traitent cependant de systèmes hétérogènes et constituent donc les tranches en fonction des propriétés réparties de façon hétérogènes dans la population, comme la bande passante ou le puissance

de calcul disponible des différents agents. Nous considérons en revanche des systèmes d'agents identiques et nous concentrons sur la taille des tranches indépendamment de la répartition individuelle des agents dans les tranches. De plus, les algorithmes de slicing considèrent souvent des agents puissants, capables, par exemple, de générer des nombres aléatoires [36], dotés d'une mémoire proportionnelle à la taille de la population [30], ou de leur voisinage dans le réseau [26]. De telles considérations sont hors de portée des agents à faible mémoire considérés dans nos modèles.

Les agents d'un protocole de population ne contrôlant pas leur propre mouvement, leur voisinage dans le réseau, constitué des agents a portée de communication change en permanence, constituant un réseau opportuniste. On retrouve de telles propriétés dans le réseau physique du projet ZebraNet [37], lequel est constitué de capteurs attachés à des zèbres en liberté dans une réserve naturelle afin de pouvoir suivre leur mouvement, des connexions sporadiques et opportunistes entre agents ponctuellement proches les uns des autres sont ensuite utilisées pour récupérer les données. Les capteurs du projet ZebraNet, bien que visant à minimiser leur consommation énergétique, restent néanmoins relativement puissants par rapport à la taille de la population de zèbres alors que nous considérons un modèle dans lequel des agents faibles peuvent être déployés dans une population de taille arbitrairement grande. D'autres projets d'étude de réseaux opportunistes étudient l'étude des réseaux de communication horizontales (ou pair-à-pair) entres terminaux mobiles transportés par leurs utilisateurs humains, tels des téléphones portables ou des PDAs. [18, 35]. Ces exemples, ne sont bien évidemment pas exhaustifs.

Contributions

Les contributions principales de ce document sont divisés en deux parties thématiques. La première partie compare les protocoles de population et les dynamiques de jeux, tandis que la deuxième partie se concentre sur les dynamiques de protocoles de populations dont la taille tend vers l'infini. Ces deux parties sont précédées d'un chapitre préliminaire (Chapitre 1) résumant l'état de l'art sur les protocoles de population. Sans prétention d'exhaustivité, ce chapitre ne présente pas seulement le modèle et les résultats initiaux d'Angluin *et al.* , mais aussi les nombreuses variantes engendrées dans les années suivantes pour tenir compte de caractéristiques aussi variées que la présence d'agents byzantins, l'auto-stabilisation, une puissance de calcul individuelle plus grande des agents ou un graphe d'interactions restreint.

La Partie I présente l'étude des similarités entre protocoles de populations et les dynamiques d'agents s'affrontant de manière répétées dans un contexte de théorie des jeux. Nous déterminons quels protocoles peuvent être vus comme le résultat d'une population d'agents s'affrontant par paires dans un jeu à deux joueurs. Chaque interaction entre deux agents est analysée comme un affrontement entre deux joueurs et la transition est vue comme chaque joueur mettant à jour sa stratégie en fonction du résultat atteint lors de la dernière manche en suivant le comportement dit de *PAVLOV*, ou *WIN-STAY, LOSE-SHIFT* ("Si je gagne, je reste. Si je perds, je change.") [46, 9]. Ce comportement est celui d'agents qui changent de stratégie lorsque leur gain est inférieur à un seuil de satisfaction, adoptant alors

mécaniquement la meilleure stratégie possible contre leur dernier adversaire.

Après un bref rappel d'éléments de théorie des jeux, le Chapitre 2 présente la manière de dériver un protocole de population de n'importe quel jeu en forme normale grâce au comportement de *PAVLOV*. Un protocole pouvant être obtenu d'un jeu de cette façon est dit *pavlovien*. Le reste du chapitre s'attache à déterminer la classe des prédicats calculables par des protocoles pavloviens. Nous montrons que limiter les protocoles de population à ceux correspondant au comportement de *PAVLOV* dans un jeu ne restreint pas leur puissance de calcul (du moins dans le cas du calcul de prédicats), puisque n'importe quel prédicat semi-linéaire reste calculable par un protocole pavlovien.

Dans le Chapitre 3, nous considérons l'effet d'une restriction des protocoles pavloviens à ceux correspondant à des jeux symétriques. En d'autres termes, nous ne considérons que les jeux dans lesquels le résultat de l'affrontement de deux stratégies est indépendant de savoir quelle stratégie est jouée par le premier joueur et laquelle est jouée par le deuxième. Nous démontrons d'abord que la contrainte de symétrie dans un protocole de population (à savoir, exiger que si la paire (a, b) devient (c, d), alors (b, a) doit nécessairement devenir (d, c)) n'est pas une véritable restriction puisque n'importe quel protocole de population peut être simulé à l'aide d'un protocole de population symétrique. En revanche, dans le cas des jeux, les protocoles pavloviens symétriques sont bien plus faibles puisqu'ils ne peuvent même pas déterminer si trois agents au moins ont commencé avec un symbole d'entrée donné. Nous proposons alors une variante, intitulée protocoles pavloviens exclusifs (*exclusive Pavlovian protocols*) qui force les agents insatisfaits à changer d'état, même si leur stratégie précédente était déjà optimale contre leur dernier adversaire. Bien que respectant toujours la contrainte de symétrie, les protocoles pavloviens exclusifs sont plus puissants, puisqu'ils sont capables de détecter au moins 3 occurrences d'un symbole. Ils sont même capables de détecter 2^k symboles pour n'importe quel $k > 1$. Malheureusement, nous n'avons pas pu déterminer de caractérisation exacte des prédicats calculables dans ce nouveau modèle.

La deuxième partie de ce document traite de très grandes populations. De telles populations, souvent approximées par des populations infines, sont un élément clef d'un grand nombre de modèles de populations, tels ceux issus des dynamiques évolutionnaires de jeux [49] ou les systèmes proie-prédateurs. Ces considérations de grandes populations sont une extension naturelle de l'hypothèse d'indépendance d'un protocole de population vis-à-vis de la taille de la population.

Le premier chapitre de cette partie s'attache à définir un modèle formle pour les grandes populations que nous appelons *Large Population Protocols* (LPP, ou protocoles de grandes population) dans lequel un grand nombre d'agents, des automates finis, interagissent par paires choisies par un ordonnanceur aléatoire suivant une probabilité uniforme. Par soucis de cohérence avec d'autres modèles de populations, nous mesurons l'évolution de la proportion d'agents dans un sous-ensemble d'états dits *marqués* au sein d'une population de taille n. Nous analysons d'abord un example simple pour montrer comment approcher l'évolution des proportions d'agents à l'aide d'une équation différentielle qui représente l'évolution d'une population infinie virtuelle. Ensuite, nous généralisons ces résultats à l'ensemble des protocoles. Ceci nous permet de définir un nombre calculé par un LPP comme étant un équilibre

exponentiellement stables de l'équation différentielle associée et de montrer que, si ν est un tel équilibre, alors la proportion d'agents dans un état marqué converge probablement vers ν quand le temps et la taille de la population tendent vers l'infini. Il est même possible d'aller plus loin et d'affirmer qu'un équilibre ν peut être approché à ϵ près avec une probabilité au moins $\mu > 0$ avec une population de taille inversement proportionnelle en ϵ et au bout d'un temps également inversement proportionnel en ϵ.

Le chapitre 5 s'attache alors à déterminer exactement quels nombres peuvent être calculés dans ce modèle. Nous montrons d'abord qu'un nombre calculable doit nécessairement être un réel algébrique dans $[0, 1]$ à l'aide d'arguments issus de la théorie des modèles. La preuve de la réciproque, que tout algébrique de $[0, 1]$ est calculable est plus complexe et ne se base pas sur de telles abstractions. Nous commençons par prouver que les nombres rationnels sont calculables puis nous montrons que tout protocole à transitions probabilistes dont les coefficients sont rationnels peut être simulé par un LPP grâce à une technique de dérandomisation constructive. Ceci revient essentiellement à prouver que l'aléa présent dans l'ordonnancement des interactions est d'une puissance suffisante pour permettre de simuler n'importe quelle distribution de probabilité calculable (c'est à dire dont les coefficients sont calculables par un LPP). Ensuite nous montrons comment construire un LPP probabiliste (et donc, grâce à la dérandomisation, un LPP déterministe) qui admette un algébrique donné comme équilibre puis nous indiquons comment il est possible de forcer la stabilité de cet équilibre. Ceci nous permet de conclure que les nombres calculables par des LPPs sont exactement les nombres réels algébriques inclus dans $[0, 1]$.

Le dernier chapitre de cette thèse est une conclusion qui inclut des pistes d'extensions de ces travaux.

Introduction

In this dissertation, we discuss models of computation in *opportunistic* networks of tiny mobile agents capable of wireless communication with pairwise interactions. Such networks are characterized by the sporadic appearance of connections between what we call *passively mobile* agents, that is to say agents that are (1) not in control of their own movement but are instead moved by a force outside of their –or their designer's– control, and (2) able to interact with other agents passing nearby, exchanging information over some undetermined wireless communication protocol. Another key property of these agents is that they are programmed identically. As a result, agents are completely anonymous and undistinguishable to an outside observer, differing only by the small amount of data they currently store, including an individual input taken from a finite set of possible inputs.

More precisely, we will consider the model of *population protocols* introduced by Angluin, Aspnes, Diamadi, Fischer, and Peralta in [2] to capture the specifics of some sensor networks. We chose this model because of its minimalistic assumptions on the computational power of the agents as well as their lack of control over the communication network. Agents in a population protocol are finite automata with wireless communication capabilities that have no unique identifiers, no control over their movement, and are identically programmed over a population of unknown finite size. They communicate with other agents on a pairwise basis, without any kind of shared memory, but simply by informing each other of their current states and updating these states accordingly. Two agents with the same state are indistinguishable to an outside observer. Computations are carried out by ensuring that the population eventually stabilizes to a set of configurations in which a desired property is preserved. For example, one could want all agents to agree on an output value, or a fixed fraction of the population to be in a given marked state.

Context and Motivations

This thesis is motivated by the general context of theoretical models dealing with the dynamics of populations. Such models have been introduced from a variety of fields of science, ranging from biology to game theory and economics, frequently overlapping fields. The Lotka-Volterra equations, for example, were introduced by Alfred J. Lotka in the context of periodic chemical reactions [41] and later used to model predator-prey systems [40]. A presentation of several such model from natural sciences and their mathematical analysis can be found in [45].

Population dynamics have also been a major field of investigation in game theory, especially in the area of evolutionary game theory [49, 33], which studies the evolution of a population of agents are playing a game against each-other repeatedly, and adapting their strategy in order to maximise some individual utility value. These dynamics have been shown to be related to, and often capture, population dynamics originating from other fields. The aforementioned Lotka-Voltera equations, for example, have been shown to be equivalent to replicator dynamics from evolutionary games [34] in which agents change their strategy by considering the difference between their individual result and the average result (or fitness) of the population.

Much of the literature on population dynamics focuses on the analytical study of different models and different dynamic systems, aiming at determining if a given dynamic eventually converges to some stable equilibrium or result in periodic patterns. In the spirit of computation however, we focus on the possibility of *programming* a population of agents so that the ensuing dynamic behaves as desired. For example, we aim at programming the agents to ensure that the population converges to a global configuration in which all agents agree on an output value depending on the initial input configuration of the population or that a predetermined fraction of agents will be in a desired marked state. This work can be seen as an extension of the model of population protocols introduced by Angluin *et al.* in [2], and an overview of the related results on population protocols can be found in chapter 1. It is known that a subclass of protocols designed in the context of evolutionary game theory correspond to Lotka Volterra dynamics [21]. We will study more generally how games can be constructed to compute any predicate computable by a population protocol using the specific *PAVLOV* dynamic. We also consider how large populations can be programmed to compute real numbers by having the population stabilize to configurations in which the ratio of agents in a given subset of states is equal to the desired number.

The problem of forcing a fixed fraction of agents to be marked can be related to the slicing problem [36] in which a population is to be split in several slices according to some predetermined rule. Most results on slicing, however, consider heterogeneous systems and try to slice according to such properties heterogeneously distributed in the population as varying computational power, bandwidth. Instead, we consider systems consisting of identical agents and focus on the size of the slices rather than which agents are set in which slice. Slicing algorithms in the literature also often consider peer-to-peer networks of powerful agents,capable, for example, of generating random numbers [36], store data of size proportional to the amount of agents in the entire population [30] or their neighbourhood [26] which are out of reach of the comparatively weak agents considered in our model.

Because the agents in a population protocol are not in control of their own movement, the set of other agents within communication range is always changing, creating an opportunistic network. This is similar to the physical ZebraNet project [37] in which sensors are attached to zebras moving around a wildlife preserve to track their movement: opportunistic wireless communications between the agents are used to collect data. In the ZebraNet project however, though minimal energy consumption is a goal, the individual agents are still relatively powerful computation-wise compared to the size of the population whereas

our models assume weak agents that can be deployed in arbitrarily large populations. Other projects studying opportunistic networks include horizontal communication between mobile devices carried by people, such as mobile phones or PDAs [18, 35]. These examples do not, of course, claim to be exhaustive.

Contributions

The main contributions of this dissertation are divided in two thematic parts. The first part compares population protocols to dynamics of games, while the second part focuses on the dynamics of population protocols when the population size grows to infinity. These two parts are preceded by a preliminary chapter (Chapter 1) summarizing the state of the art on population protocols. Without claiming to be exhaustive, this chapter presents not only the original model and results of Angluin *et al.*, but also the many variants spawned in the following years to address matters as varied as Byzantine agents, self-stabilization, enhanced individual computational power of the agents, or restricted interaction patterns.

In Part I we study the similarities between population protocols and the dynamics of populations of agents playing against each other in the context of game theory. We discuss what protocols can be considered to correspond to the decision processes of agents playing against each other repeatedly in a two-player game. Each interaction between two agents is viewed as the pair playing a game against each-other, and the transition models the players updating their strategy as a function on the result achieved in the latest encounter following the well-known *PAVLOV* or *WIN-STAY, LOSE-SHIFT* behaviour [46, 9]. In this behaviour, dissatisfied agents (that achieved less than a predetermined threshold) change their strategy for the best possible strategy to play against their latest opponent.

After first presenting some basic elements of game theory, Chapter 2 examines how a population protocol can be derived from any game in normal form via the *PAVLOV* process. We call *Pavlovian* any population protocol that can be obtained from a game in this way. The rest of the chapter then considers what kind of predicates can be computed by such Pavlovian protocols. We prove that restricting population protocols to those that correspond to the *PAVLOV* behaviour in games does not restrict the computational power of the model (at least as far as predicates are concerned), and that any semilinear predicate is also computable by a Pavlovian protocol.

Chapter 3 in turn discusses the impact of restricting Pavlovian protocols to those corresponding to symmetric games, or, in other words, to those in which the end result of two strategies being opposed is not dependent on which is played by the Initiator and which is played by the Responder. We first show that for population protocols, requiring symmetry in the transitions (that is, if the pair (a, b) updates to (c, d) then (b, a) must update to (d, c)) is not an actual restriction since any given population protocol can be simulated by an adequately constructed symmetric protocol. For games, however, we show that symmetric Pavlovian protocols are much weaker, and are, in fact, not even able to detect accurately if 3 or more occurrences of a symbol are present in the initial population. We then propose a variant called exclusive Pavlovian protocols in which dissatisfied agents are forced to change

strategy, even if their previous unsatisfactory strategy was already optimal against the latest opponent. While still symmetric, exclusive Pavlovian protocols are more powerful as they are able to detect 3 occurrences of a symbol. They are even able to detect up to 2^k occurrences for any $k \geq 1$. Unfortunately we were not able to provide a complete characterisation of the predicates computable in this new model.

In the second part of this dissertation, we deal with very large populations. Such populations, often approximated by infinite populations, are central to a great number of population models such as those of evolutionary game dynamics [49] or predator-prey systems. Such large systems come as a natural extension of the independence to size of population protocols.

The first chapter of the part focuses on the definition of a model for very large populations called *Large Population Protocols* (LPP) in which a large number of finite-state agents interact in pairs according to a uniformly random scheduler. In accordance with other population models, we consider the evolution of the proportion of agents in subset of states *called marked* states in a population of size n. First, we analyse a toy example to illustrate how the evolution of the proportion of agents in each state can be approximated by the solution of an ordinary differential equation representing the evolution of a virtual infinite population. Next, we generalize the result to general protocols. This allows us to define a number computed by a protocol to be an exponentially stable equilibrium of the corresponding differential equation, and to show that, if ν is such an equilibrium, then the fraction of agents in marked states converges probabilistically to ν when both time and population size grow to infinity. Going further, we show that such an equilibrium ν can be approximated within a margin of ϵ with probability $\mu > 0$ in a population of size and after a time both inversely polynomial in ϵ.

Chapter 5 is then naturally dedicated to characterising which numbers can be computed in the above sense. We first show that any such computable number must be an algebraic real in $[0, 1]$, using arguments from model theory. The converse proof that any algebraic real in $[0, 1]$ is computable is more complex and does not follow such high-level abstract arguments. First we prove that rational numbers are computable. Then we prove that any protocol using probabilistic transition rules in which the probabilities are rational can be simulated by a LPP using a constructive derandomisation mechanism which essentially proves that the underlying randomness of the scheduler is strong enough to simulate any computable (in the above sense) probability distribution. Then we show how to construct a probabilistic LPP (and thanks to derandomisation a deterministic LPP) accepting any given algebraic number as an equilibrium, and we explain how this equilibrium can be forced to be exponentially stable. We thereby prove that the numbers computable by LPPs are exactly the algebraic reals in $[0, 1]$.

The final chapter of this thesis is a conclusion that includes directions for future work.

Chapter 1

Population Protocols

Population Protocols were introduced by Angluin *et al.* ([2]) as a theoretical model aiming at capturing the behaviour of networks of interacting resource-limited passively-mobile agents. Agents are identically programmed mobile finite automata that are capable of wireless communication, interacting in pairs with other neighbouring agents as they come across one-another. The agents, however, do not control their movements and thus have no control over their changing neighbourhood. Computations are then made by having each agent exchange his state information with other agents passing by, and updating states according to some transition rules. A key assumption is that both the memory size and the programming are fixed, and they depend only on the problem one to be solved, and not on the population which will be running the protocol.

The canonical example introduced in [2] was that of a flock of birds in which a small fever sensor has been attached to each bird. Each fever detector is a simple machine able to detect if the bird it is attached to has higher-than-usual body temperature. This information will be considered its input for the computation. Additionally the fever detectors are assumed to have a limited computational power, represented by a fixed-sized finite memory (or set of states). They also have the ability for wireless communication with other sensors of other birds passing close by and exchange state information with them.. The programming of a sensor allows it to update its state when it exchanges information with another sensor as a function of the pair of interacting states. The goal of such a system is to detect potential epidemics, that is, if more than a certain threshold of birds have elevated temperature, in a fully distributed fashion: after some (undetermined) computation time, an outside observer should only have to check any single sensor to know whether or not there is an epidemic in the entire population.

It is easy to see that if the threshold is some fixed number of diseased birds, say 10, then using a simple counter capable to go up to 10, coupled with an alert state, the computation can be achieved by having a counter starting at 1 if the bird carrying the sensor has temperature, and at 0 otherwise. During interactions, one counter is emptied into the other, and both go into alert mode if either counter reaches 10 (or either sensor is already in alert mode). Then if the rate of encounters is sufficient to insure information dissemination, every sensor will eventually go into alert mode if 10 or more birds are sick. Is it possible to apply the

same approach if the threshold is not a fixed number, but a fixed ratio of the population? If for example an epidemic is detected when one in ten birds or more have temperature? Population Protocols bring an answer to this question by allowing us to determine which question can be anwered.

The seminal work of Angluin *et al.* [2] defines the formal model of population protocols and lays out the principle of computation, notably of predicates, using this model. It also shows how to compute predicates definable in Presburger arithmetic with population protocols on a complete interaction graph. Article [3] proves that any predicate computable in such a population protocol is, reciprocally, definable in Presburger arithmetics. While these results give a complete characterisation of the computational power of population protocols, they have spawned a large body of work on variations of the formal model, including other forms of computation with similarities with other computational models. While [7] provides a thorough and quite clear survey of the early work on population protocols, and remains a recommended introductory read on the matter, this chapter will recall many of the results presented there as well as additional later work.

1.1 Population protocols and their computational power

1.1.1 Definitions

Definition 1 (Population Protocol, [2]) *A population protocol is a* 6*-tuple* $(Q, \Sigma, Y, \iota, \omega, \delta)$ *where*

- Q *is a finite set of possible states for an agent,*
- Σ *is a finite input alphabet,*
- Y *is a (finite) set of possible outputs.*
- ι *is an input map from* Σ *to* Q*, where* $\iota(\sigma)$ *represents the initial state of an agent whose input is* σ*,*
- ω *an output map from* Q *to the output range* Y*, where* $\omega(q)$ *represents the output value of an agent in state* q*,*
- $\delta \subseteq Q^4$*, a transition relation that describes how pairs of agents can interact and update their states.*

In a population of $n \geq 2$ agents, a configuration C of the system is the vector of all the agents' states. If C is a configuration on a given population, $C(u)$ denotes the state of agent u in configuration C. Because the agents are assumed to be entirely anonymous and exchangeable, two configurations C and C' that are identical up to a permutation of the states of some or all agents are indistinguishable. Therefore a configuration can be reduced to a multiset of states or to a vector of $\mathbb{N}^Q$, representing the number of occurrence

of each possible state in the configuration. Each agent initially possesses an input value from Σ and its initial state is determined by applying ι to the initial value. This provides a population with its initial configuration. As the initial configuration is indistinguishable from any configuration that differs only by a permutation of the agents' states, inputs can similarly be reduced to a vector of $\mathbb{N}^{\Sigma}$.

A configuration C' is said to be reachable in one step from a configuration C, if there exist two agents u, v in the population such that $(C(u), C(v), C'(u), C'(v)) \in \delta$, and for any other agent w in the population, $C'(w) = C(w)$. In other words there is a pair of agents u, v that could interact to move the population from configuration C to C'. We then write $C \to C'$ to say that C' can be reached from C in one step. The transitive closure of the $\to$ relation, denoted $\to^*$, means that a configuration C' is reachable in an arbitrary number of steps from C or, in other words, that there is a sequence $C \to C_1 \to \ldots \to C_k \to C'$ of arbitrary length. A computation is then an infinite sequence of configurations $(C_k)_{k\in\mathbb{N}}$ such that for any integer $k \in \mathbb{N}$, we have $C_k \to C_{k+1}$.

Definition 2 (Fair computation) *A computation $(C_k)_{k\in\mathbb{N}}$ is said to be* fair *if, for any configurations C and C' such that $C \to C'$, if C appears infinitely often, then C' must also appear infinitely often.*

Equivalently, a computation is fair if any configuration that is always reachable will eventually be reached.

1.1.2 Computational power.

Definition 3 (Stable Computation) *A protocol $(Q, \Sigma, Y, \iota, \omega, \delta)$ is said to stably compute a function $f : \mathbb{N}^{\Sigma} \to Y$ if, for any population of any size n, and for any input vector $x \in \mathbb{N}^{\Sigma}$, any fair computation starting from the associated initial configuration eventually stabilizes to a subset of configurations in which all agents agree on a same output $y \in Y$ with $y = f(x)$.*

In the case of a binary output alphabet $Y = \{0, 1\}$, the function f can be interpreted as a predicate on the count of different input symbols in the configuration. One of the main early results on population protocols is a complete characterisation of the predicates that can be stably computed by a population protocol.

Theorem 1 ([2], [3]) *The predicates stably computable by population protocols are exactly the semilinear predicates.*

A semilinear predicate is a predicate over $\mathbb{N}^k$ for some k that can be written as a boolean combination of predicates of the form:

- $\Sigma_{i=1}^{k} a_i x_i \geq c$ where c and all a_is are (integer) constants;
- $\Sigma_{i=1}^{k} a_i x_i \equiv c(\pmod b)$ where c, b and all a_is are (integer) constants.

Such predicates are called semilinear because they are the characteristic predicates of semilinear subsets of $\mathbb{N}^k$, that is subsets of $\mathbb{N}^k$ that can be written as a finite union of sets of the from $\{\mathbf{u} + \lambda_1 \mathbf{u_1} + \ldots + \lambda_m \mathbf{u_m}, (\lambda_1, \ldots, \lambda_m) \in \mathbb{N}^m\}$, where $u_1, \ldots, u_m$ are fixed vectors of $\mathbb{N}^k$. They are known ([47]) to be exactly those predicates that can be defined in Presburger arithmetics.

In fact, [2] provides a constructive mechanism to design a population protocol corresponding to any given semilinear predicate described by its decomposition as a boolean combination of the above type. This construction relies on a mechanism to elect a leader: every agent starts with a "leader" token and one token is destroyed whenever two token-wielding agents meet. The leader is then used to compute the final result by meeting all other agents, and spreading the correct output. Angluin *et al.* [3] proved that, reciprocally, any computable predicate must be semilinear.

The characterization of computable predicates is sufficient to characterize other functions computed by population protocols. Indeed, if the (assumed finite) output alphabet Y is not binary, then, for any computable function f, and any given output $y \in Y$, the predicate f_y defined such that $f_y(x)$ is true if and only if $f(x) = y$ is computable (by a simply composing the output function ω with the characteristic function of $\{y\}$). Reciprocally, if for some function $f : \mathbb{N}^k \to Y$, all predicates f_y are computable for all $y \in Y$, then by simply running all of the corresponding protocols in parallel, and taking as output the single value y for which the corresponding predicates stably converges to output 1 one can compute f. Thus a function $f : \mathbb{N}^k \to Y$ is computable if and only if all the associated predicates $f_y, y \in Y$ are semilinear.

A large number of variations of the model have been developed after the seminal work of Angluin *et al.* Some variations increase the individual computational power of the agents or restrict the interaction pattern, others study the impact of a uniformly random scheduler, or the tolerance to faulty or Byzantine agents.

1.2 Random Scheduler

The idea of having the interacting pair of agents decided not by a fair scheduler but by picking them amongst all possible pairs of agents uniformly at random is very natural and was introduced by Angluin *et al.* [2] as a way to increase the power of population protocols. Indeed a computation decided by such a random scheduler will be fair with probability 1 and, thus, be able to compute any semilinear predicate with probability 1. But having a randomized scheduler also allows to determine the time to converge. Indeed, a fair scheduler may still delay a particularly critical interaction between two agents by any arbitrary number of rounds, whereas in a random scheduler, such an interaction always has probability $\frac{1}{n^2}$ to happen and thus will happen within an expected time of n^2. In fact, Angluin *et al.* [2] prove that any predicate can be computed with probability 1 in expected total number of rounds $O(n^2 \log n)$.

On the other hand, relaxing the probability of success might allow more advanced computation to be achieved with high probability. Indeed, Angluin *et al.* [2] show how a leader

may simulate a clock-like system by issuing a single mark and, after waiting to meet the marked agent k times, assume that $\theta(n^k)$ time has passed since the mark was issued. Using this, it can, for example, assume to have met all other agents at least once with high probability. This allows to use the non-leader agents as a finite number of unary counters which can simulate a register machine with an $O(\log n)$ bit memory [43] using time polynomial in n.

Assuming the additional prerequisite of one agent being singled out as a leader from the start, Angluin, Aspnes, and Eisenstat [4] prove that such computations can be sped up to use only $O(n \log^5 n)$ time. First is designed a language of atomic operation on the registries represented by the non-leader agents, each of which can be executed in time $O(n \log n)$.

Using the unique leader to initiate a phase clock then allows to determine when an atomic operation over the registries has been executed with high probability using the spread of a finite number m of epidemics over the population in time $\Theta(n \log n)$ to determine clock phases. This can then be used by the leader to determine with high probability when one atomic operation has had enough time to complete, thus allowing to start the next operation of a sequential program and simulate the virtual register machine. Angluin, Aspnes, and Eisenstat show how to use this virtual register machine to compute the addition, subtraction, multiplication or division using at worst $O(\log^4 n)$ atomic operations, meaning that the actual computation time for division, for example, would be $O(n \log^5 n)$, while the other operations use fewer atomic operations.

It is also shown in [4] how to use this to simulate a randomized LOGSPACE Turing machine with a constant number of read-only unary input tapes in $dn \log^2 n$ interactions with a probability of failure bounded by n^{-c} for any constant $c > 0$. Note that this brings to total computation time lower (for n large enough) than the time required to elect the leader by the destruction of initially universally distributed leader tokens.

1.3 Restricted Interactions

There are mainly two ways of restricting interactions in a population protocol and they have radically different impacts on the computational power of the protocols considered. The first is to restrict the actual interaction mechanism by only allowing one-way communication, that is by having only one agent (called the *receiver*) get the full state of the other agent (called the *sender*). The other way is to only allow certain pairs of agents to interact by defining a communication graph over the set of agents, the edges of which define legal interaction pairs (the classical model then corresponds to the complete communication graph).

1.3.1 One-way interaction

Angluin *et al.* [5] proposed several possibilities for restricting the interaction to a one-way communication. In the *observational* model the sender is not even aware that a interaction is happening (and thus cannot update its state) while in the *transmission* model, it is asked to transmit its data but has no additional information on the receiver. In addition, either

model can be further altered by adding a possible delay in the exchange of information. In this case, instead of direct interactions between agents, there are two types of actions: *send* in which a sender sends a new message which is added to a set (an unordered queue) of messages and *receive* in which a receiver takes a message from the queue and updates its state accordingly. In this *delayed* model, there is again the distinction between transmission in which the sender can update its state in a *send* action (for example to update a counter of messages sent) or an observation model in which it cannot. The receiver, of course is always able to update its state to take new information into account.

In the delayed observation model, the only computable predicates are boolean combination of predicates of the form $x_\sigma \geq 1$ that detect whether or not an input symbol σ is present in the configuration. If an additional restriction that agents may not receive their own messages is considered, then predicates of form $x_\sigma \geq 2$ detecting multiple occurrences of a symbol become computable too.

In the immediate observation model all predicates of form $x_\sigma \geq k$ for any threshold k become computable and any boolean combination thereof. Again no other predicates may be computed.

The immediate and delayed transmission model (in which the sender is aware that it is sending a message) were proven to have the same computational power, which is a strict subset of semilinear predicates but also strictly larger than the simple threshold predicates considered above as it also includes modulo predicates.

Finally if agents are allowed to refuse to receive messages when in certain states, called the *queued transmission* model (to signify that senders are aware that they are sending messages), then any two-way interaction protocol can be simulated by a queued transmission protocol.

1.3.2 Restricted communication graphs

Restricting the communication graph, that is, only allowing certain pairs of agents to interact does not reduce the computational power of population protocols as long as the graph of permitted interactions remains connected. It is possible to simulate a protocol on the complete graph by using a smaller connected graph by having neighbours swap their states to simulate movement in the graph. Contrariwise, if additional information on the structure of the communication graph is known beforehand, the designer of a protocol can take advantage of this knowledge to perform computations that would be impossible on the complete graph. Angluin *et al.* [2] proved that a straight line graph allows the simulation of a linear-space Turing machine by having each agent represent one square of the tape of the Turing machine.

If, however the graph is not known beforehand, population protocols can be used to decide properties of the underlying graph. A population protocol can be used, for example (see [1]), to determine if the underlying graph of communications of a given population has maximum degree less than some predefined finite value k. Angluin *et al.* [1] give several other types of properties of graphs that can be computed by using population protocols.

1.4 Enhanced individual computational power

A section of work on population protocols has focused on the impact of increasing the individual power of agents in a population protocol, generally in keeping with the idea that the individual computational power should remain small compared to the size of the population.

1.4.1 Unique identifiers

Gerraoui and Ruppert [31] proposed a model, called *community protocols* in which the state-space of the agents is extended with a an identifier storing capacity (of size $O(\log n)$). Each agent starts with a unique identifier and can store a fixed finite number of other identifiers. The only operations allowed on these identifier are testing for equality and, after an interaction, the set of identifiers stored by the two interacting agents should be a subset of the set of identifiers known to either of them before the interaction.

Such a community protocol can then compute a predicate if and only if that same predicate can be computed by a Turing machine with $O(n \ \log n)$ space and permutation of the input characters does not affect the output value. Since a population of n agents with any kind of memory of size $O(\log n)$ could be simulated using a Turing machine with $O(n \ \log n)$, this means that restricting the use of the extra memory to identifier storage (with the addition of individual unique identifiers at start) is not restrictive when compared to general purpose $O(\log n)$ memory.

1.4.2 Passively mobile machines

Going further, Chatzigiannakis *et al.* [20] consider using communication-capable Turing Machines as agents instead of finite automata. Calling their model *Passively mobile machines,* their main result is that a predicate is computable by passively mobile machines using $O(f(n))$ space with $f(n) = \Omega(\log \ n)$ if and only if it is computable by a Non-deterministic Turing Machine with $O(nf(n))$ space and the output does not change if a permutation of the input characters occurs.

Additionally, Chazigiannakis *et al.* proved that if the memory size used by passively mobile machines is $f(n) = o(\log \ \log \ n)$, then the predicates computable are exactly the semilinear predicates, just as in the classical population protocol model.

1.5 Fault tolerance

Fault tolerance, that is the ability of a distributed system to compute a correct output despite faulty operation by some of the components of the system, is a major component of the literature on distributed system and it was only to be expected that population protocols would be studied to determine such fault tolerance properties. In general, faults are divided in several categories : crashes in which an agent ceases to operate completely, transient

failures which corrupt the memory of a single agents arbitrarily altering its current state and Byzantine failures in which an agent does not behave according to specification.

1.5.1 Byzantine Agents

The exact behaviour of a Byzantine agent may vary from simple unreliability of the state to, in the worst case scenario, malignant design of an agent actively trying to undermine the computation by any means possible (e.g. not interacting accordingly with the fairness property, giving false information or not updating its state properly), possibly even with information about the entire system that a regular agent could not possibly have. With such vast possibilities for disrupting computation it should come as no surprise that Byzantine agents can wreak havoc among simple interacting computing machines such as are found in population protocol. Indeed, Guerraoui and Ruppert [31] proved that in the presence of even a single Byzantine agent, only trivial predicates (that is predicates that are universally true or universally false) can be computed. They proved, however that the community protocol model, enhancing traditional population protocol with unique identifiers, allowed for some level of resistance to Byzantine agents as long as the Byzantine were not allowed to alter identifiers. In this case, community protocols can compute any decision problem in $NSPACE(n \log n)$ in the presence of $f = O(1)$ Byzantine agents, given preconditions ensuring the output does not change if up to f of the input characters are changed and up to $3f + 1$ of the input characters are deleted.

This means that, in addition to faking their own input (which cannot be prevented), f Byzantine agents can, in a sense, prevent up to $3f + 1$ agents to be acknowledged by the rest of the population.

1.5.2 Crashes and Transient Failures

On the front of Crash failures, in which an agent simply stops operating completely, Delporte-Gallet *et al.* [23] proved that, in a similar fashion, if a function f is computable in a failure-free environment and if $\mathcal{D}$ is a subset of all possible input multisets such that taking any input multiset $I \in \mathcal{D}$ and removing $c + t$ input symbols and adding t, then f can be computed in an environment tolerating up to c crashes and t transient failures as long as the input I is in $\mathcal{D}$ if the population is sufficiently large.

In other words, c crashes and t transient failures can be tolerated without loss of computational power if the additional precondition is respected that removing c input symbols and altering t more does not change the output value. A better margin cannot be hoped since, the $c + t$ alterations (or removals) correspond to possible crashes and failures that would occur before the computation has even started in which case no recovery would be possible. Note that the method described in [23] requires the population to be larger than $2((|Y| + 2)(c + 2t) + 2)^2$ where $|Y|$ is the number of output symbols. An algorithm to turn any protocol into such a fault-tolerant protocol is given and the resulting protocol requires $O(|Y|(c+2t))$ memory and is thus dependant on the fixed number of faults to be determined (transient failures being, in a sense, twice as bad a crashes).

1.6 Self Stabilization

The problem of making population protocols self-stabilizing, that is to have a population protocol able to achieve a desired property even if starting from a bad configuration, has been investigated in several papers [27, 6]. If, in [6], Angluin *et al.* show how to make some protocols presented in [2] self-stabilizing, such as distance-2 colouring in a bounded-degree graph, ring orientation on a distance-2 coloured graph, or the computation of a spanning tree in a regular graph of bounded degree D.[1]

The leader election protocol, so central to the computation of semilinear predicates in [2] is however proven to be impossible to achieve in population protocols on general graphs in a self-stabilizing fashion but a protocol is given for the special case of odd-sized rings. Fischer and Jiang [27] proved that an eventual leader detector oracle (that eventually informs the agents if no leader is present in the population) is sufficient to design a self-stabilizing leader election protocol on the complete graph or on the ring.

Finally, Beauquier *et al.* [12, 11] introduced the idea of having a *Base Station* with unlimited capacity with which the agents may also interact and use this to design self-stabilizing protocols. [12] shows that even with a base station, the problem of counting the number of (regular) agents is impossible to solve if the agents' memory size is smaller than the number of agents. In [11], introduce the additional assumption of the existence cover times for each agents, that is, a bound on the time a given agent may have to wait to be sure to have met all other agents. Such cover times are used to model differences in speed or mobility pattern for the agents in the population. Agents are unaware of their cover time but are able, when meeting, to determine which of them is the faster agent (ie. the agent with the lower covering time). Using the base station to start the computation and, subsequently to initiate non overlapping phases in which the protocol is repeatedly executed, Beauquier *et al.* show how to construct a self-stabilizing protocol from any non self-stabilizing such protocols. It is also shown that this non overlapping repeated execution would be impossible without the base station. In [10], Beauquier *et al.* show how the base station and covering times model can be used to compute the Gathering problem, in which input values distributed in the population must be transferred to the base station exactly once.

[1] The construction of the spanning tree would actually work for unbounded degree graphs but requires $O(logD)$ memory for each agent.

Part I

Games and Population Protocols

Pairwise interactions between finite-state agents are motivated by the study of the dynamics of particular two-player games in game theory. Indeed, the characteristics of population protocols make them an efficient framework to describe the dynamics of agents repeatedly playing a game against each other. In this part of the thesis, we aim at better understanding when pairwise interactions, and, more specifically, population protocols, can be considered as the result of a game. We turn two-player games into dynamics over agents by considering *PAVLOV* behaviour (sometimes also termed *WIN-STAY, LOSE-SHIFT* [9, 46]). This is inspired by [25, 28, 38] that consider the dynamics of a particular set of rules termed the *PAVLOV* behaviour in the iterated Prisoners' Dilemma. Notice that we will extended the *PAVLOV* behaviour from two-strategy two-player games to n-strategy two-player games, whereas above references only consider two-strategies two-player games (in fact, mostly of the iterated Prisoners' Dilemma).

The *PAVLOV* behaviour is not the only way to associate a dynamic to a game. Alternatives to *PAVLOV* behaviour could include *MYOPIC* dynamics (at each step each player chooses the best response to previously played strategy by its adversary, regardless of result), or the well-known and well-studied *FICTIOUS-PLAYER* dynamics [17] (at each step each player chooses the best response to the statistics of the past history of strategies played by its opponents). The reader is refered to, e.g., [13, 29] for a presentation of known results about the properties of dynamics resulting from some properties of the underlying game. Also, [9] presents a zoology of possible behaviours for the iterated Prisoners' Dilemma game, with discussions of their compared merits.Another way to associate a dynamic to a game is using the *Replicator* Dynamic (see [34]) in which a population of agents repeatedly play against each other (pairwise). Each agent updates his strategy according to a probability distribution depending on the individual payout received in an iteration, and on global results achieved by the different possible strategies achieved in the population. Other dynamics associated to games are descriped in [49] from the point of view of Evolutionary Games Theory.

The first chapter of this part will introduce a new way in which a Population Protocol can be associated to a game, using the *PAVLOV* dynamic. After giving a few structural properties on the protocols that may be obtained from a game in such a way, protocols that we logically name Pavlovian Population Protocols (PPP), we show that all predicates computable by Population Protocols can be computed by a protocol corresponding to a game, i.e. by a PPP. The second chapter discusses the restriction to symmetric games. This chapter does not present a characterisation of the computational power of Population Protocol but should instead be viewed as a discussion of the restrictive nature of symmetric games. Two results, however, stand out. The first is that symmetric Population Protocol, that is Population Protocols in which interactions are independent on the order of the interacting pair, without consideration for games, are as powerful as general Population Protocol. Indeed, we show that any Population Protocol can be simulated using a symmetric Population Protocol. The second is that, contrariwise, restricting Pavlovian Population Protocols to symmetric games (and thus, symmetric interaction rules) is a strong restriction: no Symmetric Population Protocol can compute the simple threshold predicate "are there three or more agents in a given state." We then introduce a slight change in the *PAVLOV* behaviour, which we call

Exclusive PAVLOV Behaviour that allows us to circumvent this limitation. It is unclear, however, if Exclusive Pavlovian Protocols are as powerful as Population Protocols as the predicates we actually know how to compute are far more limited, and rely heavily on the pairwise nature of the interactions.

Chapter 2

Pavlovian Population Protocols

In this chapter, study a new way to associate a population protocol via the *PAVLOV* (or *WIN-STAY, LOSE-SHIFT* [9]) behaviour. After recalling some elements of game theory, we show how a population protocol can be derived from a game in normal form using the *PAVLOV* behaviour. Calling any protocol that can be obtained from a game this way a Pavlovian population protocol (or PPP), we then focus on the computational power of PPPs, showing that any predicate computable by a population protocol, that is any semilinear predicate, can also be computed by a Pavlovian population protocol. Similarly to the method used in [2], we achieve this by showing how to construct Pavlovian population protocols to compute each of the building blocks of semilinear predicates. Part of the results described in this chapter were published in [14].

2.1 Elementary Game Theory

Let us first recall the simplest concepts from Game Theory which will be useful in this chapter. We focus on non-cooperative two-player games, with complete information, in normal form. That is, we focus on games made up of two players, called I (or *initiator*) and R (or *responder*), with a finite set of actions, called *pure strategies*, $Strat(I)$ and $Strat(R)$. $A_{i,j}$ and $B_{i,j}$ respectively denote the score for player I and R, when I uses strategy $i \in Strat(I)$ and R uses strategy $j \in Strat(R)$. The scores are given by $n \times m$ matrices A and B, where n and m are the cardinalities of $Strat(I)$ and $Strat(R)$.

Example 1 (Prisoner dilemma) *The Prisoner dilemma, is a game inspired the following scenario: two prisoners, accused of committing a crime together are being questioned separately by the police. Each of them is faced with the following choice: either cooperate with their partner and deny everything, hoping that limited evidence will prevent a harsh conviction or aggressively accuse their partner to minimize their own conviction by shifting most of the blame on him.*

- *If both prisoners keep denying, they face one year in jail due to the lack of evidence for a stronger conviction.*

- *If one accuses the other who denies, a bargain deal is offered to the accuser in which he will walk free in exchange for his testimony which will allow his partner to be convicted to six years in jail.*
- *If both accuse each other, the prosecution can get a conviction for three years for each.*

In normal form, this game can be described by the following matrices, denoting by C (for cooperation) the first pure strategy, and by D (for defection) the second pure strategy of each player

$$A = \begin{pmatrix} -1 & -6 \\ 0 & -3 \end{pmatrix}, B = \begin{pmatrix} -1 & 0 \\ -6 & -3 \end{pmatrix}$$

(The results are counted negatively to show that a higher result is better).

Note that this game is symmetric, that is $B =^t A$, illustrating that a player's result is the same, whether he acts as initiator or responder.

A strategy x in $Strat(I)$ is said to be a *best response* to strategy y in $Strat(R)$, denoted by $x \in BR_A(y)$ if $A_{z,y} \leq A_{x,y}$ for all strategies $z \in Strat(I)$. Conversely, a strategy $y \in Strat(R)$ satisfies $y \in BR_B(x)$ if $B_{x,z} \leq B_{x,y}$ for all strategies $z \in Strat(R)$. A pair (x, y) is a *(pure) Nash equilibrium* if $x \in BR_A(y)$ and $y \in BR_B(x)$. In other words, two strategies (x, y) form a Nash equilibrium if in that state neither of the players has a unilateral interest to deviate from it.

There are two main approaches to discuss dynamics of games. The first consists in repeating games [13], while the second consists in using models from evolutionary game theory. We refer to [34, 49] for a presentation of this latter approach. Repeating a given game $\mathcal{G}$ a fixed (and known by the players) number of times k can be considered to be a single game $\mathcal{G}^{(k)}$ in which a strategy for each player is defined as the succession of strategies she picks in the individual iterations of $\mathcal{G}$. If the strategies for $\mathcal{G}$ are respectively $Strat(I)$ and $Strat(R)$, then, a strategy for $\mathcal{G}^{(k)}$ would be to give a strategy of $\mathcal{G}$ for each of the iterations of $\mathcal{G}$ in $\mathcal{G}^{(k)}$. This strategy may be depending on the previous iterations and on their result so the final number of strategies for $\mathcal{G}^{(k)}$ is equal to $n(nm)^{k-1}$ (resp. $m(nm)^{k-1}$) for player I (resp. R), assuming that both players are assumed to keep the same role in each iteration, where n and m are the respective number of strategies for I and R in $\mathcal{G}$

If a game $\mathcal{G}$ is iterated an infinite (or unknown) number of times, the problem cannot be described in the same manner. In practice, however, player I (respectively R) still has to solve the following problem at each time t: given the history of the game up to t, that is, given $X_{t-1} = x(1), \cdots, x(t-1)$ and $Y_{t-1} = y(1), \cdots, y(t-1)$ what should I(resp. R) play at time t? In other words, how to choose $x(t) \in Strat(I)$ (resp. $y(t) \in Strat(R)$), given X_{t-1} and Y_{t-1}.

In this context, it is natural to suppose that the answer to this problem is given by some behaviour rules of the form $x(t) = f(X_{t-1}, Y_{t-1})$ and $y(t) = g(X_{t-1}, Y_{t-1})$ for some particular functions f and g.

The question of finding the best behaviour rule to use in games, in particular for the Prisoners' Dilemma gave birth to an important literature. The book [9] describes the results

of tournaments of behaviour rules for the iterated Prisoners' Dilemma. It argues that there exists a best behaviour rule called *TIT-FOR-TAT*. This rule consists in cooperating at the first step, and then do the same thing as the adversary at subsequent step. A lot of other behaviours, most of them with very picturesque names, have been proposed and studied in [9]. Among possible behaviours, is the *PAVLOV* behaviour [9, 38, 46]: in the iterated Prisoners' Dilemma, a player cooperates if and only if both players opted for the same strategy in the previous move. The name *PAVLOV* stems from the fact that this strategy embodies an almost reflex-like response to the pay-off: a player repeats her former action if she was rewarded above a threshold value, and switches behaviour if it was punished by receiving under this value. We refer to [46] for some study of this strategy in the spirit of Axelrod's tournaments. The *PAVLOV* behaviour can also be termed *WIN-STAY, LOSE-SHIFT* because if the play in the previous step resulted in a success, then the agent will play the same strategy in the next step, while if the play resulted in a failure, then the agent switches to another action [9, 46].

According to [34], "evolutionary games dynamic is the application of population dynamical methods to game theory." This is leading us to the study of a population of agents playing a game repeatedly against each other, and adapting their strategies depending on their pay-off. This approach is similar to a Population Protocol in which the states are the strategies, and the transitions represent changes of strategies after playing one round of the game. We assume that, in the underlying game, the set of strategies for player I is the same as for player R, as is the case in the Prisoner Dilemma.

2.2 From Games To Population Protocols

In the spirit of the discussion in ethe previous section, we can associate a Population Protocol to any game as follows. Our association is corresponding to a *PAVLOV*(ian) behaviour. Recall that a Population Protocol is defined by its set Q of possible states, the set $\delta \subset Q^4$ of transition rules, an input function and an output function.

Definition 4 (Associating a Protocol to a Game) *Assume a (possibly asymmetric) two-player game $\mathcal{G}$ is given. Let A and B be the gain matrices of player Iand R, respectively and let Δ be some threshold.*

A protocol associated to game $\mathcal{G}$ is a population protocol whose set of states is $Q = Strat(I) = Strat(R)$ and whose set δ of transition rules satisfies $(q_1, q_2, q_1', q_2') \in \delta$ if and only if

$$q_1' = \begin{cases} q_1 \text{ if} A_{q_1,q_2} \geq \Delta, \\ x \in BR_A(q_2) \text{ otherwise} \end{cases} \quad \text{and } q_2' = \begin{cases} q_2 \text{ if} B_{q_2,q_1} \geq \Delta, \\ x \in BR_B(q_1) \text{ otherwise.} \end{cases}$$

Several Population Protocols can be associated to the same game, differing only from their input and output functions, or from the threshold Δ considered.

Definition 5 (Pavlovian Population Protocol) *A Population Protocol is* Pavlovian *if it can be obtained from a game by the rules of Definition 4.*

A Pavlovian population protocol will be termed *deterministic* if best responses are assumed to be unique. In this case, the rules are deterministic: for all q_1, q_2, there is a unique q_1' and a unique q_2' such that $(q_1, q_2, q_1', q_2') \in \delta$.

In order to avoid systematically talking about matrices, we are now stating some structural properties of deterministic Pavlovian Population Protocols.

Consider a set of rules. For all rules $ab \to a'b'$, let us denote $\delta_a^I(b) = b'$ and $\delta_b^R(a) = a'$. Let $\text{Stable}^I(a) = \{x \in Q | \delta_a^I(x) = x\}$, and $\text{Stable}^R(a) = \{x \in Q | \delta_a^R(x) = x\}$.

Lemma 1 *A set of rules is deterministic Pavlovian iff for all states $q \in Q$ there exist $\max^I(q) \in \text{Stable}^I(q)$ and $\max^R(q) \in \text{Stable}^R(q)$ such that, for all states a and b,*

1. $b \notin \text{Stable}^I(a) \Rightarrow \delta_a^I(b) = max^I(a)$,
2. $b \notin \text{Stable}^R(a) \Rightarrow \delta_a^R(b) = max^R(a)$.

Proof. First, we consider a deterministic Pavlovian population protocol $\mathcal{P}$ obtained from a game with pay-off matrices A and B. Let Δ be the threshold yielding the aforementioned protocol according to Definition 4. Let a be an arbitrary state in Q, and let q be the best response to strategy a for matrix B.

Le us focus on the rule $aq \to a'q'$ where $(a', q') \in Q^2$, i.e., we focus on the case where player I plays a while player R plays q. As $q = BR_B(a)$, we have, by Definition 4, $q' = q$. Thus, $q \in \text{Stable}^I(a)$.

Now, let us consider b such that $b \notin \text{Stable}^I(a)$. We focus on the rule $ab \to a''b'$ where $(a'', b') \in Q^2$. By definition of set Stable^I, we have $b \neq b'$. Using Definition 4, we have $B_{b,a} < \Delta$ and $b' = BR_B(a)$. So $b' = BR_B(a) = q$. Thus, if we let $max^I(a) = q$, then $max^I(a)$ satisfies the conditions of the proposition.

Using similar arguments, we can also prove that there exists $max^R(a) \in \text{Stable}^R(a)$ such that $b \notin \text{Stable}^R(a)$ implies $\delta_a^I(b) = max^R(a)$. In fact, we can sum up the relationship between the game matrix and rules as follows: for any $a \in Q$, we have

$$\text{Stable}^I(a) = \{x \in Q | B_{x,a} \geq \Delta\} \cup \{BR_B(a)\},$$

$$max^I(a) = BR_B(a),$$

$$\text{Stable}^R(a) = \{x \in Q | A_{x,a} \geq \Delta\} \cup \{BR_A(a)\},$$

$$max^R(a) = BR_A(a).$$

Conversely, consider a Population Protocol $\mathcal{P}$ satisfying the properties of the proposition. All rules $ab \to a'b'$ are such that $\delta_a^I(b) = b'$ and $\delta_b^R(a) = a'$. We focus on the construction on a two-player game having pay-off matrices A and B. We fix an arbitrary value Δ as the threshold of the corresponding game.

- If $\text{Stable}^I(a) \neq Q$, then $B_{max^I(a),a} = \Delta + 1$. If $x \in \text{Stable}^I(a)$ and $x \neq max^I(a)$, then $B_{x,a} = \Delta$. If $x \notin \text{Stable}^I(a)$, then $B_{x,a} = \Delta - 1$.
- If $\text{Stable}^I(a) = Q$, then $\forall x \in Q$, $B_{x,a} = \Delta$.
- If $\text{Stable}^R(a) \neq Q$, then $A_{max^R(a),a} = \Delta + 1$. If $x \in \text{Stable}^R(a)$ and $x \neq max^R(a)$, then $A_{x,a} = \Delta$. If $x \notin \text{Stable}^R(a)$, then $A_{x,a} = \Delta - 1$.
- If $\text{Stable}^R(a) = Q$, then $\forall x \in Q$, $A_{x,a} = \Delta$.

This game describes all rules of $\mathcal{P}$. Thus, $\mathcal{P}$ is a Pavlovian population protocol. □

2.3 Main Result

Given a Pavlovian population protocol computing a given predicate Ψ, we can simply invert the binary output function to obtain a Pavlovian population protocol computing the negation of Ψ. The class of predicates computable by a Pavlovian population protocol is therefore closed under negation. However, it is not clear that predicates computable by Pavlovian population protocols are closed under conjunction or disjunction. This becomes true if one considers *multi-protocols*. The idea is to consider k (possibly asymmetric) two-player games. At each step, each player chooses a strategy for each of the k games and each of the k games is played independently when two agents meet. Formally:

Definition 6 (Multi-protocol) *Consider k (possibly asymmetric) two-player games. For game i, let Q^i be the corresponding states, and A^i and B^i the corresponding matrices.*

The associated Population Protocol is the Population Protocol whose set of states is $Q = Q^1 \times Q^2 \times \ldots \times Q^k$, and whose transition rules are given as follows:

$$((q_1^1, \ldots, q_1^k), (q_2^1, \ldots, q_2^k), (q_1^{1'}, \ldots, q_1^{k'}), (q_2^{1'}, \ldots, q_1^{k'})) \in \delta$$

if and only if $(q_1^i, q_2^i, q_1^{i'}, q_2^{i'})$ is a transition of the Pavlovian population protocol associated to the ith game, for all $1 \leq i \leq k$.

Notice that multi-protocols are just a particular kind of population protocols. This is the key property used in [2] to prove that stably computable predicates are closed under boolean operations. When considering Pavlovian games, one can build multi-protocols that are not Pavlovian protocols, and it is not clear whether one can always transform any Pavlovian multi-protocol into an equivalent Pavlovian protocol.

We consider predicates ψ over vectors of non-negative integers. We write $[\psi]$ for their characteristic functions. Recall that a predicate is semi-linear if and only if it is Presburger definable [47]. Semi-linear predicates correspond to boolean combinations of threshold predicates and modulo predicates defined as follows (variables x_i represent the number of agents initially in state σ_i): A *threshold* predicate is of the form $[\Sigma a_i x_i \geq k]$, where for any $i, a_i \in \mathbb{Z}$,

$k \in \mathbb{Z}$, and the x_i are variables. A *modulo* predicate is of the form $[\Sigma a_i x_i \equiv b \mod k]$, where for any $i, a_i \in \mathbb{Z}$, $k \in \mathbb{N} \setminus \{0, 1\}$, $b \in [1, k-1]$, and the x_i are variables.

We can then state our main result:

Theorem 2 *For any predicate ψ, the following conditions are equivalent:*

- *ψ is computable by a Population Protocol ;*
- *ψ is computable by a Pavlovian population multi-protocol ;*
- *ψ is semi-linear.*

The proof of this theorem is simply the conjunction of Lemma 2, giving us closure under boolean operations and Lemma 3 and 4 that give us both types of building blocks enabling to compute semi-linear predicates (respectively threshold predicates and modulo predicates). That no-more than semi-linear predicates are computable is, of course, the main result of [3] on general population protocols applied to the smaller class of Pavlovian population protocols.

We now prove that our multi-games give us closure under boolean operations. Then, in the following sections, we will prove that threshold predicates and modulo predicates are indeed computable.

Lemma 2 *The class of predicates computable by multi-games is closed under boolean operations.*

Proof. The negation is easier to deal with, as we just need to change the output function ω into $1 - \omega$. From de Morgan's laws, we then only need to prove closure by conjunction. For the conjunction of two multi-games, let consider the multi-game including all the games present in the two initial multi-games. The conjunction protocol is the one associated to the new multi-game with output function

$$\omega'(q^1, \ldots, q^k, p^1, \ldots, p^l) = \omega_1(q^1, \ldots, q^k) * \omega_2(p^1, \ldots, p^l),$$

where $(q^1, \ldots, q^k)$ and $(p^1, \ldots, p^l)$ are the corresponding games in the first and second multi-games, and ω_1 and ω_2 the respective output functions. $\square$

Since from Lemma 2, predicates computable by Pavlovian population multi-protocols are closed under boolean operations, and since a Pavlovian population protocol is a particular Pavlovian population multi-protocol, and since predicates computable by (general) population protocols are known to be exactly semi-linear predicates, to prove Theorem 2 we only need to prove that we can compute threshold and modulo predicates by Pavlovian population protocols. This is the purpose of the following sections.

2.4 Threshold Predicates

In this section, we prove that we can compute threshold predicates using Pavlovian protocols.

Lemma 3 *For any integer k, and any integers $a_1, a_2, \cdots, a_m$ there exists a Pavlovian population protocol that computes $[\sum_{i=1}^{m} a_i x_i \geq k]$.*

First note, that we can assume without loss of generality that $k \geq 1$. Indeed,

$$[\Sigma a_i x_i \geq -k] = [\Sigma(-a_i)x_i \leq k] = [\Sigma(-a_i)x_i < k+1]$$

which is the negation of $[\Sigma(-a_i)x_i \geq k+1]$. Thus from a Population Protocol computing $[\Sigma(-a_i)x_i \geq k+1]$ with $k \geq 0$, we just have to inverse the output function to obtain a Population Protocol that computes $[\Sigma a_i x_i \geq -k]$.

The purpose of the rest of this section is to prove Lemma 3. We first discuss some basic ideas. Our techniques are inspired by the work of Angluin et al. [5]. The set of states we use is the set of integers from $[-M, M]$ where $M = \max(|a_i|, 2k-1)$. Each agent with input σ_i is given an initial weight of a_i. During the execution, the sum of the weights over the whole population is preserved. In [3], the general idea is the following: two interacting agents with positive weights p and q such that $p + q \leq M$ are transformed into an agent with weight 0 and an agent with weight $p + q$, while two agents with weight p and q such that $p + q > M$ are transformed into two agents with respective weight $\lfloor (p+q)/2 \rfloor$ and $\lceil (p+q)/2 \rceil$ which are both greater or equal to k.

In our setting, we cannot use the same transformation as above since all agents that change their states when they meet an agent in state p while being initiator (resp. responder) must take the same state, which only depends on p. To avoid this problem, a trick is to use rules of the following form: $pq \to (p+1)(q-1)$. However, we also have to make sure that the protocol enables all agents to agree in the final configuration. While is easy in the classical population protocol model, this turns out to be more complicated in our setting.

We describe a protocol that computes $[\sum_{i=1}^{m} a_i x_i \geq k]$. The protocol is defined as follows: we consider

$$\Sigma = \{\sigma_1, \ldots, \sigma_l\},$$

$$Q = \{\top\} \cup [-M, M].$$

Moreover, for all i, $\iota(\sigma_i) = a_i$/ Finally, we set $\omega(\top) = 1$ and, for any $p \in [-M, M]$, $\omega(p) = 1$ if and only if $p \geq 1$.

We distinguish two cases: either $k = 1$, or $k \geq 2$. We present two protocols, because we need to have a mechanism enabling us to "broadcast" the result. This is not so difficult in the case where $k = 1$, whereas it is more technical if $k \geq 2$.

Case $\mathbf{k = 1}$. Our protocol computing $[\Sigma a_i x_i \geq 1]$ is defined as follows. The rules are the following.

$$
\begin{array}{lll}
\top\top & \to \top\top & \\
\top x & \to \top x & \text{when } x \in [-M, M] \\
1\top & \to 1\top & \\
1n & \to (n+1)\top & \text{when } n \in [-M, 0] \\
1p & \to 1p & \text{when } p \in [1, M]
\end{array}
$$

For all, $n \in [-M, 0]$, and all $p \in [2, M-1]$,

$$
\begin{array}{lll}
n\top & \to n0 & \\
nx & \to nx & \text{when } x \in [-M, M] \\
p\top & \to p\top & \\
pn & \to (n+1)(p-1) & \\
pp' & \to pp' & \text{when } p' \in [1, M].
\end{array}
$$

Case k $\geq$ 2. Our protocol is deterministic and, from Lemma 1, uniquely determined by the sets $\text{Stable}^I(q)$ and $\text{Stable}^R(q)$, and by the values $max^I(q)$ and $max^R(q)$ defined as follows.

$q \in Q$	$\text{Stable}^I(q)$	$max^I(q)$	$\text{Stable}^R(q)$	$max^R(q)$
$\top$	$\{\top\} \cup [-M, 0] \cup [k, M]$	-1	$\{\top\} \cup [-M, M]$	0
$n \in [-M, -1]$	$[-M, M]$	0	$\{\top\} \cup [-M, 0]$	$(n+1)$
0	$[-M, M]$	0	$\{\top\} \cup [-M, k-1]$	1
1	$\{\top, 0, M\}$	$\top$	$[-M, 0]$	2
$p \in [2, k-1]$	$\{\top, 0, M\}$	$(p-1)$	$[-M, 0]$	$(p+1)$
$b \in [k, M-1]$	$\{\top\} \cup [k, M]$	$(b-1)$	$\{\top\} \cup [-M, 0] \cup [k, M]$	$(b+1)$
M	$\{\top\} \cup [k, M]$	$(M-1)$	$\{\top\} \cup [-M, M]$	M

The transition rules we obtain from these sets and values are the following.

$$
\begin{array}{lll}
\top\top & \to \top\top & \\
\top x & \to \top x & \text{when } x \in [-M, 0] \cup [k, M] \\
\top p & \to (p+1)(-1) & \text{when } p \in [1, k-1] \\
1\top & \to 1\top & \\
10 & \to 10 & \\
1M & \to 1M & \\
1x & \to (x+1)\top & \text{when } x \notin \{\top, 0, M\}
\end{array}
$$

For all $p \in [2, k-1]$,

$$
\begin{array}{lll}
p\top & \to p\top & \\
p0 & \to p0 & \\
pM & \to pM & \\
px & \to (x+1)(p-1) & \text{when } x \notin \{\top, 0, M\}
\end{array}
$$

For all $n \in [-M, 0]$,

$$
\begin{array}{lll}
n\top & \to n0 & \\
nx & \to nx & \text{when } x \in [-M, M]
\end{array}
$$

For all $b \in [k, M]$,

$$\begin{array}{lll} b\top & \to b\top & \\ bx & \to (x+1)(b-1) & \text{when } x \in [-M, k-1] \\ bb' & \to bb' & \text{when } b' \in [k, M] \end{array}$$

We say that an agent in state $x \in [-M, M]$ has weight x and that an agent in state $\top$ has weight 0. Note that in the initial configuration the sum of the weights of all agents is exactly $\Sigma a_i x_i$. Note also that any of the rules of our protocol does not modify the total weight of the population, i.e., at any step of the execution, the sum of the weights of all agents is exactly $\Sigma a_i x_i$.

The stable configurations, (i.e., the configurations where no rule can be applied to modify the state of any agent), are the following:

- every agent a is in some state $n(a) \in [-M, 0]$,
- a unique agent is in state $p \in [1, k-1]$ and every other agent is in state 0.
- every agent a is either in some state $b(a) \in [k, M]$ or in state $\top$.

Note that no agent starts in state $\top$, and that no rule enables the two interacting agents to enter the state $\top$ except for the rule $\top\top \to \top\top$. Thus, we know that it is impossible that all agents are in state $\top$. Consequently, in the last stable configuration, there is at least one agent in a state $b \in [k, M]$. Moreover, in any stable configuration, all agents have the same output. If $\Sigma a_i x_i \geq k$ then all agents output 1, while in all the other cases, the agents output 0. Thus, if the population reaches a stable configuration, then the computed output is correct and it will not be modified any more. Now, we should prove that the fairness condition ensures that the population always reaches a stable configuration. In fact, it is sufficient to prove that, from any reachable configuration, there exists an execution that reaches a stable configuration.

Consider any configuration reached during the execution. As long as there is an agent in state $p \in [1, M]$ and an agent in state $n \in [-M, -1]$, we apply $pn \to (n+1)(p-1)$. Thus we can always reach a configuration where the states of all agents are in $[-M, 0] \cup \{\top\}$ if $\Sigma a_i x_i \leq 0$ and in $[0, M] \cup \{\top\}$ otherwise.

If $\Sigma a_i x_i \leq 0$, then there is at least one agent in state $n \in [-M, 0]$, because the agents cannot all be in state $\top$. In this case, applying iteratively the rule $n\top \to n0$, we reach a stable configuration where all agents have a state in $[-M, 0]$.

Suppose now that $\Sigma a_i x_i \in [1, k-1]$. Since $\Sigma a_i x_i \in [1, k-1]$, each agent with a positive weight is in a state in $[1, k-1]$. Applying iteratively the rule $pp' \to (p-1)(p'+1)$ where $p, p' \in [1, k-1]$, we reach a configuration where there is exactly one agent in state $p \in [1, k-1]$ while all the other agents are in state 0 or $\top$. Applying iteratively the rules $\top p \to (p+1)(-1)$ and $(p+1)(-1) \to 0p$, we reach a configuration where one agent is in state $p \in [1, k-1]$ while all the other agents are in state 0.

Finally, assume that $\Sigma a_i x_i \geq k$. If there is an agent in state $p \in [1, k-1]$, then there is at least another agent in state $q \in [1, M]$. If $p + q \leq M$, then applying iteratively the

rule $pq \to (p-1)(q+1)$ between these two agents, we reach a configuration where one of these two agents is in state 0 while the other is in state $p+q$. In this case, we have strictly reduced the number of agents with states in $[1, k-1]$. If $p+q > M \geq 2k$, then $q \in [k, M]$, and applying iteratively the rule $qp \to (q-1)(p+1)$, we reach a configuration where one agent is in state k while the other agent is in state $p+q-k \in [k, 2M]$. Here again, we have strictly reduced the number of agents with states in $[1, k-1]$. Applying these rules as long as there exists an agent in state $p \in [1, k-1]$, we reach a configuration where every agent is either in a state in $[k, M]$, or in state 0 or $\top$. Since $\Sigma a_i x_i \in [k, M]$,there exists at least one agent in state $b \in [k, M]$. Applying iteratively the rules $b0 \to 1(b-1)$ and $1(b-1) \to b\top$, we reach a stable configuration where all agents are either in state $\top$ or in a state in $[k, M]$.

2.5 Modulo Counting

The aim of this section is to prove Lemma 4 which is the second building bloc that we use to prove Theorem 2.

Lemma 4 *For any two integers $k \geq 2$ and b, and for any integers $a_1, a_2, \cdots, a_m$, there exists a Pavlovian population protocol that computes $[\sum_{i=1}^{m} a_i x_i \equiv b \mod k]$.*

We treat separately the cases $b = 0, k > 2$ and $b \neq 0, k \geq 2$. The case $b = 0, k = 2$ is deduced by considering that $[\sum_{i=1}^{m} a_i x_i \equiv 0 \mod 2]$ is the negation of $[\sum_{i=1}^{m} a_i x_i \equiv 1 \mod 2]$.

Claim 1 *For any integer $k > 2$ and any family of integers $a_1, a_2, \cdots, a_m$ there exists a Pavlovian Population Protocol that computes $[\sum_{i=1}^{m} a_i x_i \equiv 0 \mod k]$.*

Proof. The protocol is defined as follows:

- $\Sigma = \{\sigma_1, \ldots, \sigma_l\}$.
- $Q = \{A, B\} \cup [0, k-1]$, where A and B correspond to weight 0 but with different output meaning than state 0.
- $\iota(\sigma_i) \equiv a_i \mod k$.
- $\omega(0) = 1$ and for any $x \in [1, k-1] \cup \{A, B\}$, $\omega(x) = 0$.

Our protocol is deterministic and, from Lemma 1 uniquely determined by the sets $\text{Stable}^I(q)$, $\text{Stable}^R(q)$, and values $max^I(q)$, $max^R(q)$ defined as follows.

$q \in Q$	$\text{Stable}^I(q)$	$max^I(q)$	$\text{Stable}^R(q)$	$max^R(q)$
A	$Q \setminus \{B\}$	0	$Q \setminus \{B\}$	0
B	$Q \setminus \{A\}$	0	$Q \setminus \{A\}$	0
0	$\{A, B, 0, 1, (k-1)\}$	$(k-1)$	$[0, k-1]$	0
1	$\{A\}$	A	$Q \setminus \{1\}$	2
$(k-1)$	$Q \setminus \{(k-1)\}$	$(k-2)$	$\{B\}$	B
$p \in [2, k-2]$	$\{A, B\} \cup [0, p-1]$	$(p-1)$	$\{A, B\} \cup [p+1, k-1]$	$(p+1)$

The transition rules we obtain from these sets and values are the following.

$$
\begin{array}{rcll}
AA & \rightarrow & AA & \\
AB & \rightarrow & 00 & \\
A0 & \rightarrow & 00 & \\
Ap & \rightarrow & Ap & \text{when } p \in [1, k-2] \\
A(k-1) & \rightarrow & B(k-1) & \\
BA & \rightarrow & 00 & \\
BB & \rightarrow & BB & \\
B0 & \rightarrow & 00 & \\
Bp & \rightarrow & Bp & \text{when } p \in [1, k-1] \\
0x & \rightarrow & 0x & \text{when } x \in \{1, B, 0, 1\} \\
Op & \rightarrow & (p+1)(k-1) & \text{when } p \in [2, k-2] \\
0(k-1) & \rightarrow & B(k-1) & \\
1x & \rightarrow & 1A & \text{when } x \in \{A, B, 0\} \\
1p & \rightarrow & (p+1)1 & \text{when } p \in [1, k-2] \\
1(k-1) & \rightarrow & BA &
\end{array}
$$

and, for all $p \in [2, k-2]$,

$$
\begin{array}{rcll}
px & \rightarrow & px & \text{when } x \in \{A, B\} \cup [0, p-1] \\
pn & \rightarrow & pn & \text{when } n \in [p, k-2] \\
p(k-1) & \rightarrow & B(p-1) &
\end{array}
$$

In the initial configuration, the sum of the weights of all agents is exactly $\Sigma a_i x_i \mod k$. The application of any rule of the protocol does not modify this value, i.e., at any step of the execution, the sum of the weights of all agents is exactly $\Sigma a_i x_i \mod k$.

The stable configurations, (i.e., the configurations where no rules can be applied to modify the output of any agent), are the following:

- all agents are in state 0.
- there exists a unique agent in state $p \in [1, k-2]$ and all other agents are in state A.

- there exists a unique agent in state $p \in [2, k-1]$ and all other agents are in state B.

In any stable configuration, either all agents output 1 (if $\Sigma a_i x_i \equiv b \mod k$), or all agents output 0. We now show that, from any reachable configuration, we can reach a stable configuration.

We apply the same reasoning we used before. As long as there are two agents in states $p, p' \in [1, k-1]$ with $p \leq p'$, we can apply iteratively the rule $pp' \to (p'+1)(p-1)$ between these two agents. Doing so, either one agent enters state 1, or one agent enters state $k-1$. If one agent is in state 1 while the other is in state $k-1$, we apply the rule $1(k-1) \to BA$, and we have decreased the number of agents with a positive weight by 2. If there is one agent in state 1 (resp. $k-1$) while the other is in state $p \in [1, k-2]$ (resp. $p \in [2, k-1]$), then we apply the rule $1p \to (p+1)A$ (resp. $p(k-1) \to B(p-1)$) to decrease the number of agents with a positive weight. Thus, we can always reach a configuration where there is at most one agent a in state $p \in [1, k-1]$, while every other agent is in state 0, A or B.

If $\Sigma a_i x_i \equiv 0 \mod k$, then either all agents have started in state 0 or not. In the first case, the initial configuration was a stable configuration, and we have nothing to prove. In the second case, the last step before we reach a configuration containing only agents in state 0, A or B, there was one agent in state 1, one agent in state $(k-1)$ and the rule $1(k-1) \to BA$ has been applied. Thus, either there exists an agent in state 0 in the configuration, or there exists an agent in state A and an agent in state B. In the latter case, we can apply the rule $AB \to 00$ to be in a configuration containing agents in state 0. Then, applying the rules $A0 \to 00$ and $B0 \to 00$, we reach a stable configuration where all agents are in state 0.

If $\Sigma a_i x_i \equiv 1 \mod k$ (resp. $\Sigma a_i x_i \equiv k-1 \mod k$), then we reach a configuration where there is exactly one agent in state 1 (resp. $k-1$) while every other agents is in state 0, A or B. Then applying the rules $1B \to 1A$ and $10 \to 1B$ (resp. $A(k-1) \to B(k-1)$ and $0(k-1) \to B(k-1)$), we reach a stable configuration where there is exactly one agent in state 1 (resp. $k-1$) while all the other agents are in state A (resp. B).

If $\Sigma a_i x_i \equiv p \mod k$ with $p \in [2, k-2]$, we reach a configuration where there is exactly one agent in state p while all the other agents are in state 0, A or B. If the agents with weight 0 are either all in state A, or all in state B, we are in a stable configuration. Otherwise, applying the rules $AB \to 00$, $0p \to (p+1)(k-1)$ and $(p+1)(k-1) \to Bp$ as long as possible, we reach a final configuration where exactly one agent is in state p while all the other agents are in state B. □

Claim 2 *For any integers $k \geq 2, b \in [1, k-1]$, and any family of integers $a_1, a_2, \cdots, a_m$, there exists a Pavlovian population protocol that computes $[\sum_{i=1}^{m} a_i x_i \equiv b \mod k]$.*

Proof. The protocol is defined as follows:

- $\Sigma = \{\sigma_1, \ldots, \sigma_l\}$.
- $Q = \{\top\} \cup [0, k-1]$, where $\top$ corresponds to a weight of 0 but has a different output meaning than the state 0.

- $\iota(\sigma_i) \equiv a_i \mod k$.

- $\omega(\top) = 1$ and for any $p \in [0, k-1]$, $\omega(p) = 1$ if and only if $p = b$.

Our protocol is deterministic and, from Lemma 1, uniquely determined by the sets $\text{Stable}^I(q)$, $\text{Stable}^R(q)$, and by the values $max^I(q)$, $max^R(q)$ defined as follows. In the following table, when $p = k-1$ (resp. $b = k-1$), $p+1$ (resp. $b+1$) should be understood as $\top$.

$q \in Q$	$\text{Stable}^I(q)$	$max^I(q)$	$\text{Stable}^R(q)$	$max^R(q)$
$\top$	Q		$\{\top, 0, b\}$	1
0	$Q \setminus \{b\}$	$(k-1)$	$[0, k-1]$	0
b	$\{\top\} \cup [0, b-1]$	$(b-1)$	$\{\top\} \cup [b+1, k-1]$	$(b+1)$
$p \in [1, k-1] \setminus \{b\}$	$[0, p-1]$	$(p-1)$	$\{\top, 0\} \cup [p+1, k-1]$	$(p+1)$

The transition rules we obtain from these sets and values are the following.

$$
\begin{array}{rll}
\top x & \to \top x & \text{when } x \in \{\top\} \cup [0, k-1] \\
0\top & \to 0\top & \\
0b & \to \top(k-1) & \text{when } b = k-1 \\
0b & \to (b+1)(k-10 & \text{when } b \neq k-1 \\
0p & \to 0p & p \in [0, k-1] \setminus \{b\} \\
b\top & \to b\top & \\
bp & \to bp & \text{when } p \in [0, b-1] \\
bp & \to (p+1)(b-1) & \text{when } p \in [b, k-2] \\
b(k-1) & \to \top(b-1) &
\end{array}
$$

and, for all $p \in [1, k-1] \setminus \{b\}$

$$
\begin{array}{rll}
p\top & \to 1(p-1) & \\
pp' & \to pp' & \text{when } p' \in [0, p-1] \\
pp' & \to (p'+1)(p-1) & \text{when } p' \in [p, k-2] \\
p(k-1) & \to \top(p-1) &
\end{array}
$$

Again, in the initial configuration the sum of the weights of all agents is exactly $\Sigma a_i x_i \mod k$ and the application of any rule of our protocol does not modify this value, i.e., at any step of the execution, the sum of the weights of all agents is exactly $\Sigma a_i x_i \mod k$.

The stable configurations, (i.e., the configurations where no rule can be applied to modify the state of some agent), are the following:

- there exists a unique agent in state b and all other agents are in state $\top$.

- there exists at most one agent in state $p \in [1, k-1] \setminus \{b\}$ and all other agents are in state 0.

As in the protocols for threshold predicates, no rule can transform the states of two interacting agents into $\top$ except for the rule $\top\top \to \top\top$. Since no agents are initially in state $\top$, it is impossible to reach a configuration where all agents are in state $\top$.

In any stable configuration, either all agents output 1 (if $\Sigma a_i x_i \equiv b \mod k$), or all agents output 0. We now show that from any reachable configuration, we can reach a stable configuration.

As long as there are two agents in states $p, p' \in [1, k-1]$ with $p \leq p'$, we can apply iteratively the rule $pp' \to (p'+1)(p-1)$ between these two agents. Doing so, either one agent enters state 0, or one agent is in state $k-1$, while the other is in state $p'' \in [1, k-1]$. In the latter case, by applying the rule $p(k-1) \to \top(p-1)$, we decrease the number of agents with a positive weight by 1. Thus, we can always reach a configuration where there is at most one agent a in state $p \in [1, k-1]$, while every other agent is in state 0 or $\top$.

If $\Sigma a_i x_i \equiv 0 \mod k$, then every agent is in state 0 or $\top$, and there is at least one agent in state 0. Applying the rule $\top 0 \to 00$ as often as necessary, we reach a stable configuration where all agents are in state 0.

If $\Sigma a_i x_i \equiv p \mod k$ with $p \in [1, k-1] \setminus \{b\}$ then one agent is in state p while every other agent is in state 0 or $\top$. If there is an agent in state $\top$, then we apply the rule $p\top \to 1(p-1)$. If $p = 1$, then there is one more agent in state 0. If $p > 1$, we apply the rule $1(p-1) \to p0$, to also get one more agent in state 0. Iterating this process, we reach a stable configuration where one agent is in state p while all other agents are in state 0.

If $\Sigma a_i x_i \equiv b \mod k$, then one agent is in state b while all the other agents are in state 0 or $\top$. As long as there is an agent in state 0, if $b \neq k-1$, then we apply the rules $0b \to (b+1)(k-1)$ and $(b+1)(k-1) \to \top b$. If $b = k-1$, then we apply the rule $0(k-1) \to \top(k-1)$. Doing so, we reach a stable configuration where one agent is in state b while all other agents are in state $\top$. $\square$

We can now finally prove Lemma 4

Proof of Lemma 4. Claim 1 and Claim 2 combined prove that Lemma 4 holds for $k > 2$, whatever the value of b. In fact, Claim 2 proves that Lemma 4 also holds for $k = 2$ and $b = 1$. All we are left with is the special case where $k = 2$ and $b = 0$. Notice that $[\Sigma a_i x_i \equiv 0 \mod 2]$ is exactly $\not[\Sigma a_i x_i \equiv 1 \mod 2]$. Thus, by taking the protocol for $k = 2$ and $b = 1$ from Claim 2, and inverting the output function (in the same ways as in the proof of Lemma 2), we get a Pavlovian protocol computing $[\Sigma a_i x_i \equiv 0 \mod 2]$. $\square$

We now have all the building blocks to compute all semi-linear predicates, proving that Pavlovian population multi-protocols have the exact same computational power as population protocols.

2.6 Conclusion

In this chapter we introduced a natural way to derive population protocol from any two-player game in normal form (in which both players have access to the same set of strategies)

via the *PAVLOV* behaviour. Calling population protocols derived in this way Pavlovian population protocols, we showed that any predicate definable in Presburger arithmetic could be computed by a Pavlovian population multi-protocol. That is, semilinear predicates can be computed by a protocol in which several games are played in parallel by the two agents selected for an interaction. Each agent then updates his strategy for each game independently, according to the results of the *PAVLOV* behaviour for this game. Because Pavlovian multi-protocols are a subset of population protocols, this means that the predicates computable by Pavlovian multi-protocols are exactly those definable in Presburger arithmetic. This makes Pavlovian multi-protocols as powerful as population protocols to stably compute predicates.

Chapter 3

Symmetric Games and protocols

The purpose of this chapter is to discuss the power of symmetric Pavlovian protocols, that is, Pavlovian population protocols that are obtained from games in which the Initiator and the Responder play symmetric rules. Such games yield Pavlovian protocols in which the transition rules are symmetric too, meaning that the order of an interacting pair has no impact on the transition rule followed in an interaction. We first show that restricting general population protocols to symmetric ones is not truly a restriction as any population population protocol can be simulated by a population protocol with symmetric transition rules. We then show that this result does not, however hold for Pavlovian protocols as no Pavlovian protocol can detect if three or more occurrences of an input symbol are present in the population. We then show a way to circumvent this limitation by introducing *Exclusive* symmetric Pavlovian protocols, in which a dissatisfied agent is forced to change his strategy, even if his previous strategy was already the best response to his opponent's strategy. We show that exclusive symmetric Pavlovian protocols are strictly more powerful than symmetric Pavlovian protocol.

A game were the score matrix A for the *Initiator* is the transpose of the score matrix B for the *Responder* is called symmetric. The distinction between *Initiator* and *Responder* is then no longer pertinent as both play a similar role and the game can simply be described from the subjective point of view of each agent as between a *Player* and an *Opponent.*

Returning to the Example 1, the Prisoner dilemma, where A and B are the following matrices

$$A = \begin{pmatrix} -1 & -6 \\ 0 & -3 \end{pmatrix}, B = \begin{pmatrix} -1 & 0 \\ -6 & -3 \end{pmatrix},$$

as the game is symmetric, matrix A and B can also be summarized by the following (single) matrix:

		Opponent	
		C	D
Player	C	R	S
	D	T	P

in which every player confronts his own strategy to the strategy of his opponent to determine his personal score.

On the other hand, a Population Protocol is considered to be symmetric if its transitions are symmetric. That is to say, for every rule $(q_1, q_2, q_3, q_4) \in \delta$, you also have $(q_2, q_1, q_4, q_3) \in \delta$.

The Pavlovian protocol obtained from a symmetric game is also symmetric. Indeed, the position of a player as *Initiator* or *Responder* has no bearing on the scored achieved by playing a given strategy opposed to another, the *PAVLOV* response will be the same in both situations.

3.1 Symmetric population protocols

In this section, we prove that Symmetric population protocols are as powerful as general population protocol.

Definition 7 (Symmetric Population Protocol) *A population protocol is said to be symmetric if and only if the set δ of transition rules is such that $(q_1, q_2, q_3, q_4) \in \delta$ if and only if $(q_2, q_1, q_4, q_3) \in \delta$.*

Theorem 3 *If the population is of size ≥ 3, any population protocol can be simulated by a symmetric population protocol.*

Proof. The idea of the proof is to construct a symmetric Population Protocol simulating a given (asymmetric) population protocol by using two copies of the set of states from the original protocol and using them to determine wether an agent should be the first or the second agent in an asymetric interaction. The key to the proof is to have agents alternate between both roles (that is between the two copies of the set of states) when they encounter another agent trying to play the same role in the interaction. Thus when two agents trying to act as first members of a transition interact, they both move to try to act as second members of a transition the next time they are picked to interact (without changing their state with relation to the first protocol), and vice-versa. Note that we assume all protocols to be deterministic here.

Let $\mathcal{P} = (Q, \Sigma, \iota, \omega, \delta)$ be some (asymmetric) Population Protocol. We will now construct a symmetric Population Protocol $\mathcal{P}' = (Q', \Sigma, \iota', \omega', \delta')$ such that any computation in $\mathcal{P}'$ can be mapped to a computation in $\mathcal{P}$ and, conversely, any computation in $\mathcal{P}$ for a population of size ≥ 3 can be simulated in $\mathcal{P}'$. We will then say that $\mathcal{P}'$ simulates P and it is clear that $\mathcal{P}'$ computes the same predicate as $\mathcal{P}$. This, of course implies that we use the same input alphabet Σ for both protocols.

We define $Q' = Q \times \{1, 2\}$ to have two copies of Q. For any state $q \in Q$ we call an agent in state $(q, 1)$ (resp. $(q, 2)$) a *first-minded* (resp. *second-minded*) agent in state q. We use natural extensions for ι and ω: for any symbol $\sigma \in \Sigma, \iota'(\sigma) = (\iota(\sigma), 1)$ and for any state $q \in Q$, we set $\omega'(q, 1) = \omega'(q, 2) = \omega(q)$. Finally, we define δ' from δ as follows. For any pair $(u, v) \in Q$, let $(u', v') = \delta(u, v)$. Then set:

- $\delta'((u,1),(v,1)) = ((u,2),(v,2)$ and $\delta'((u,2),(v,2)) = ((u,1),(v,1))$
- $\delta'((u,1),(v,2)) = ((u',1),(v',2))$ and $\delta'((v,2),(u,1)) = ((v',2),(u',1))$.

This defines $\mathcal{P}'$ as a symmetric protocol. To see that any valid computation in $\mathcal{P}'$ corresponds to a valid computation in $\mathcal{P}$, simply project the states of all agents according to their first coordinate (that is, disregard whether they are first- or second-minded). Any computation $C_1 \to \ldots \to C_k \to \ldots$ in $\mathcal{P}'$ becomes a valid computation in $\mathcal{P}$ with a few additional stagnations in the same configuration (when two identically-minded agents change from first- to second-minded or the other way around).

Conversely, consider a possible transition $C_1 \to C_2$ in $\mathcal{P}$ where C_1 and C_2 are two configurations of a population of size $n \geq 3$. Then for any configuration C_1' of the same population of agents but over states in Q', such that we have $C_1'(w) \in C(w) \times \{1,2\}$ for any agent w in the population. (Recall that, for a configuration C and an agent $w, C(w)$ denotes the state of w in C.) That is, any configuration in which we only added a first- or second-minded state to each agent satisfies that there is a configuration C_2' reachable from C_1' such that, for any agent $w, C_2'(w) \in C_2(w) \times \{1,2\}$. Indeed, let (u,v) be two agents such that $C_1 \overset{u,v}{\to} C_2$. We have $(C_2(u), C_2(v)) = \delta(C_1(u), C_1(v))$ and, by definition of δ',

$$\delta'((C_1(u),1),(C_1(v),2)) = ((C_2(u),1),(C_2(v),2)).$$

If $C_1'(u) = (C_1(u),1)$ and $C_1'(v) = (C_1(v),2)$, then by having agents u,v interact we directly get a satisfactory configuration C_2'. Otherwise, we will show that we can, using a third agent w (hence the condition that $n \geq 3$) change the mind of agents u and v to get to this case.

1. If u,v,w are all identically-minded, then having w interact with whoever is of the wrong mind (v if they are all first-minded, u otherwise) results in the desired configuration for u and v.
2. If u,v are identically-minded but w is differently-minded, then having u,v interact brings us back to case 1.
3. If u is second-minded and v is first-minded, then having w interact with whoever is like-minded brings us back to case 1.

This proves that any valid computation in $\mathcal{P}$ can be simulated by a valid configuration in $\mathcal{P}'$ (with the addition of at most two "mind changes" at each computation step). □

The immediate corollary of Lemma 3 is that symmetric population protocols have the exact same computational power as population protocols.

Corollary 1 *Any predicate computable by a population protocol is also computable by a symmetric population protocol.*

In the case of Pavlovian Protocols however, requiring symmetry is very restrictive.

Theorem 4 *There is no symmetric Pavlovian protocol that computes the threshold predicate* $[x.\sigma \geq 3]$*, i. e., the predicates which is true when there are at least* 3 *occurrences of input symbol* σ *in the input* x*.*

Proof. The proof is by contradiction. Assume that there exists such a symmetric Pavlovian protocol $\mathcal{P}$. Without loss of generality, we may assume that $\Sigma = \{0, \sigma\}$ is a subset of the set of states Q. Let A be the gain matrix from a symmetric game associated with this protocol. In keeping with a previous remark, we may assume without loss of generality that $\Delta = 0$ is the gain threshold for the *PAVLOV* behaviour corresponding to $\mathcal{P}$. We will prove contradiction by showing that $\mathcal{P}$ cannot possibly distinguish between the inputs $x_3 = \{\sigma, \sigma\}$ and $x_4 = \{\sigma, \sigma, \sigma, \sigma\}$.

Since the protocol is symmetric, for any $q \in Q$, the rule $qq \to q'q''$, is such that $q' = q''$, that is to say of the form $qq \to q'q'$. Let us consider the sequence of rules such that $\sigma\sigma \to q_1q_1 \to q_2q_2 \to \cdots \to q_kq_k \to \ldots$ where $\sigma, q_1q_2, q_3, \ldots, q_k \in Q$. Since Q is finite, there exist two distinct integers k and ℓ such that $q_k = q_\ell$ and $k < \ell$.

The case $k + 1 = \ell$ is not possible. Indeed, we would have the rule $q_kq_k \to q_kq_k$. Then, consider the inputs $x_3 = \{\sigma, \sigma\}$ and $x_4 = \{\sigma, \sigma, \sigma, \sigma\}$. x_4 must be accepted. From x_4 there is a derivation $x_4 \to \{q_1, q_1, \sigma, \sigma\} \to \{q_1, q_1, q_1q_1\} \to^* \{q_k, q_k, q_k, q_k\}$. This latter configuration is terminal from the above rule. Since x_4 must be accepted, we must have $\omega(q_k) = 1$. However, from x_3 there is a derivation $x_3 \to \{q_1, q_1\} \to^* \{q_k, q_k\}$, where the last configuration is also terminal. We reach a contradiction, since the output of this last configuration would be $\omega(q_k) = 1$, whereas x_3 must be rejected. Hence, $k + 1 < \ell$, and $q_kq_k \to q_{k+1}q_{k+1} \to \cdots \to q_\ell q_\ell = q_kq_k$. Let T be the set of states $T = \{q_i : k \leq i \leq \ell\}$. Since $q_iq_i \to q_{i+1}q_{i+1}$ is among the rules, by definition of Pavlovian behaviour, we have $q_{i+1} = BR(q_i)$.

Let us discuss the rules

$$q_iq_j \to q'_iq'_j \tag{3.1}$$

for $(q_i, q_j) \in T^2$. There are two possibilities for the value of q'_i:

$$q'_i = \begin{cases} q'_i = q_i \text{ if } A_{q_iq_j} \geq 0 \\ q'_i = BR(q_j) = q_{j+1} \text{ otherwise.} \end{cases}$$

In any case, the value of q'_i is in T. Symmetrically, we have two possibilities for q'_j, all of them in T. Hence, all rules of the form (3.1) preserve T: we have $q'_i, q'_j \in T$, as soon as $q_i, q_j \in T$.

Similarly to what we did in the case $k + 1 = \ell$, there is a derivation

$$x_4 \to \{q_1, q_1, \sigma, \sigma\} \to \{q_1, q_1, q_1q_1\} \to^* \{q_k, q_k, q_k, q_k\}.$$

From the last configuration, by the previous remark, the state of all agents will be in T. Since x_4 must be accepted, ultimately all agents will be in states that belong to T and such that their image by ω is 1. Consider now x_3. There is a derivation

$$x_3 \to \{q_1, q_1\} \to^* \{q_k, q_k\}$$

that will go trough all configurations $\{q_i q_i\}$, for all $q_i \in T$. This cannot eventually stabilize to elements whose image by ω is 0, because some of the elements of T have image 1 by ω, and hence x_3 is not accepted. This yields the desired contradiction. $\square$

The bad news of Theorem 4 is balanced by the fact that, as we will show, a small modification of the *PAVLOV* behaviour enables us to compute the threshold predicate $[x.\sigma \geq 3]$. This small modification leads us to define *Exclusive PAVLOV* behaviour. Exclusive *PAVLOV* behaviour differs from standard *PAVLOV* behaviour in that it requires an agent to adopt a different strategy if it has achieved a negative score. This must hold even if the current strategy is already the best response to the strategy played by the Opponent. When such a case arises, the Player selects the second-best strategy, that is, the best strategy different from the current strategy.

We denote by $BR_{\neq x'}(y)$ the set of best responses to strategy y, different from strategy x. Similarly to what was done in Definition 4, we can then define an *Exclusive Pavlovian Protocol* to be a Protocol obtained from a game by following the *Exclusive PAVLOV* behaviour instead of the traditional *PAVLOV* behaviour.

Definition 8 (Exclusive Pavlovian Protocol) *Given a symmetric two-player, let A be its gain matrix and let Δ be some threshold. A protocol exclusively associated to the game is a population protocol whose set of states is $Q = Strat(I) = Strat(R)$, the set of strategies of the game, and whose transition rules δ are given by: $(q_1, q_2, q_1', q_2') \in \delta$ if and only if*

$$q_1' = \begin{cases} q_1 \text{ if } A_{q_1,q_2} \geq \Delta \\ x \in BR_{\neq q_1}(q_2) \text{ otherwise,} \end{cases}$$

$$q_2' = \begin{cases} q_2 \text{ if } B_{q_2,q_1} \geq \Delta \\ x \in BR_{neqq_2}(q_1) \text{ otherwise.} \end{cases}$$

The input and output function can be defined independently from the game.

An exclusive Pavlovian protocol is a Pavlovian protocol exclusively associated to a symmetric game.

Note that the definition above only differs from Definition 4 in that the game is assumed to be symmetric, and in the use of $BR_{\neq q_i}(q_j)$ instead of $BR(q_j)$.

Let us first show that exclusive Pavlovian protocol are, in fact, a superset of symmetric Pavlovian protocols.

Theorem 5 *Any symmetric Pavlovian protocol is also an exclusive Pavlovian protocol.*

Proof. Let $\mathcal{P}$ be a symmetric Pavlovian protocol. Let $\mathcal{G}$ be some symmetric game associated to $\mathcal{P}$ and let A be the gain matrix of $\mathcal{G}$. Without loss of generality we may assume that the corresponding gain threshold $\Delta = 0$.

Denote B the matrix defined by

$$B_{i,j} = \begin{cases} A_{i,j} \text{ if } i \notin BR(j) \\ A_{i,j} \text{ if } i \in BR(j) \text{ and } A_{i,j} \geq 0 \\ O \text{ otherwise.} \end{cases}$$

Then, if $i \in BR(j), B_{i,j} \geq 0$. Now define $\mathcal{G}'$ to be a symmetric game with gain matrix B. The best responses to any given strategy q in $\mathcal{G}'$ are the same as the best responses in $\mathcal{G}$ because the only gain changes is that of best responses and the gain of best responses is either unchanged or increased to 0. Therefore, we do not distinguish the set of best response to a strategy q in $\mathcal{G}$ and in $\mathcal{G}'$.

$\mathcal{P}$ is exclusively associated to $\mathcal{G}'$ with threshold $\Delta = 0$. Indeed, because the best response to a strategy q_2 always has a positive gain when playing agaist q_2 in $\mathcal{G}'$, we do not need to force a change in strategy if $q_1 \in BR(q_2)$. □

We are now going to describe some structural properties on exclusive Pavlovian protocols similar to those Lemma 1 describes for Pavlovian protocols. Consider a set of rules. For any rule $ax \to a'x'$, denote $\delta_a(x) = x'$. Since we consider symmetric rules, we then have symmetrically $\delta_x(a) = a'$. Let

$$\text{Stable}(a) = \{x \in Q | \delta_a(x) = x\}.$$

Lemma 5 *A set of rules is exclusive Pavlovian if and only if* $\forall a \in Q\ \exists\ max(a) \in Q$ *such that*

$$\text{Stable}(a) \neq \emptyset \Rightarrow max(a) \in \text{Stable}(a)$$

$$\forall b \notin \text{Stable}(a) : b \neq max(a) \Rightarrow delta_a(b) = max(a).$$

Proof. First, we consider an Exclusive Pavlovian Population Protocol P obtained from corresponding matrix M. Without loss of generality, we can consider the associated threshold to be $\Delta = 0$. Let a be an arbitrary state in Q, and q be the best response to strategy a for matrix M (i.e $q = BR(a)$). Then $max(a) = q$ and $\text{Stable}(a) = \{b \in Q, M_{b,a} \geq 0\}$ by definition of Exclusive Pavlov behaviour.

If $\text{Stable}(a) \neq \emptyset$, then there exists a state $b \in \text{Stable}(a)$ and $M_{b,a} \geq 0$. Thus, by definition of $q, M_{q,a} \geq M_{b,a} \geq 0$ and $q \in \text{Stable}(a)$.

Now, let us consider $b \notin \text{Stable}(a)$ (if there exists one). Consider the rule $ab \to a''b'$ where definition of set $\text{Stable}(a)$, we have $b \neq b'$. Using Definition 4, we have $M_{b,a} < 0$ and $b' = BR(a) = q$.

Conversely, consider a Population Protocol P satisfying the properties of the lemma. All rules $ab \to a'b'$ are such that $\delta_a(b) = b'$ and $\delta_b(a) = a'$. We focus on the construction on a two-player game having the corresponding matrix M. We fix 0 as the threshold of the corresponding game.

- If $\text{Stable}(a) = Q$, then $\forall x \in Q$, $M_{x,a} = 0$.

- If Stable$(a) = \emptyset$, then we have $M_{max(a),a} = -1$ and $M_{\delta_a(max(a)),a} = -2$. Moreover, $M_{x,a} = -2$ for each state x except $max(a)$,$\delta_a(max(a))$.

- If $\emptyset \subset$ Stable$(a) \subset Q$ then the value $M_{x,a}$ of each element x depends on the set Stable$(a)$and $max(a)$.

$$M_{max(a),a} = 1.$$

  If $x \in$ Stable(a) and $x \neq max(a)$, then $M_{x,a} = 0$. If $x \notin$ Stable(a), then $M_{x,a} = -1$.

This game describes all rules of P. So, P is an exclusive Pavlovian population protocol. □

As a consequence of Lemma 5 a protocol is given by describing, for any state a,

1. the set Stable(a),
2. the value of $max(a)$,
3. and whenever, Stable$(a) = \emptyset$, the value of $\delta_a(max(a))$.

Example 2 *Consider the Pavlovian Population Protocol associated to the game with following pay-off matrix :*

		Opponent						
		0	1	2_+	2_-	X	Y	$\top$
	0	1	*0*	*-3*	*-3*	*0*	*0*	-1
	1	*0*	*-1*	*-3*	*-3*	*-1*	*-1*	-1
	2_+	*-1*	*1*	*-3*	*-1*	*-1*	*-1*	0
Player	2_-	*-1*	*0*	*-1*	*-3*	*-1*	*-1*	0
	X	*0*	*0*	*-2*	*-3*	*-1*	*0*	-1
	Y	*0*	*0*	*-3*	*-2*	*0*	*-1*	-1
	$\top$	*0*	*0*	*-3*	*-3*	*1*	*1*	1

Such a protocol can be described by the parameters Stable(a), $max(a)$, $\delta_a(max(a))$ *for each state a:*

a	Stable(a)	$max(a)$	$\delta_a(max(a))$
0	$\{0, 1, X, Y, \top\}$	0	0
1	$\{0, 2_+, 2-, X, Y, \top\}$	2_+	2_+
2_+	$\emptyset$	2_-	X
2_-	$\emptyset$	2_+	Y
X	$\{0, Y, \top\}$	$\top$	$\top$
Y	$\{0, X, \top\}$	$\top$	$\top$
$\top$	$\{2_+, 2_-, \top\}$	$\top$	$\top$

Note that if Stable$(a) \neq \emptyset$, *then* $\delta_a(max(a)) = max(a)$. *Otherwise* $\delta_a(max(a)) \neq max(a)$.

3.2 Some simple exclusive Pavlovian protocols

Now that we have the structural properties of Lemma 5, we can prove that exclusive Pavlovian protocols are more powerful than simple symmetric Pavlovian protocols, in that they at least allow us to count up to three.

3.2.1 Counting up to 3 exclusively.

The following result shows that exclusive Pavlovian protocols are indeed more powerful than simple symmetric Pavlovian protocols, in that at least they can count up to three.

Proposition 1 *There is an exclusive Pavlovian protocol that computes the threshold predicate* $[x.\sigma \geq 3]$*, which is true when there are at least* 3 *occurrences of input symbol* σ *in the input* x*.*

Proof. Let us consider the following protocol :

- $Q = \{0, 1, 2_+, 2_-, X, Y, \top\}$.
- $\Sigma = \{x, y\}$.
- $\iota(x) = 1$, $\iota(y) = 0$.
- $\omega = 1_{\{\top\}}$.
- with interaction rules defined as follows (omitting transition rules of form $ab \to ab$ that leave the system unchanged):

$$
\begin{array}{c}
00 \to 00 \\
01 \to 01 \\
02_+ \to 2_-0 \\
02_- \to 2_+0 \\
0X \to 0X \\
0Y \to 0Y \\
0\top \to \top\top \\
11 \to 2_+2_+ \\
12_+ \to 2_-2_+ \\
12_- \to 2_+2_- \\
1X \to \top X \\
1Y \to \top Y \\
1\top \to \top\top \\
2_+\top \to 2_+2_- \\
2_-\top \to 2_-2_+ \\
X\top \to \top\top \\
Y\top \to \top\top
\end{array}
$$

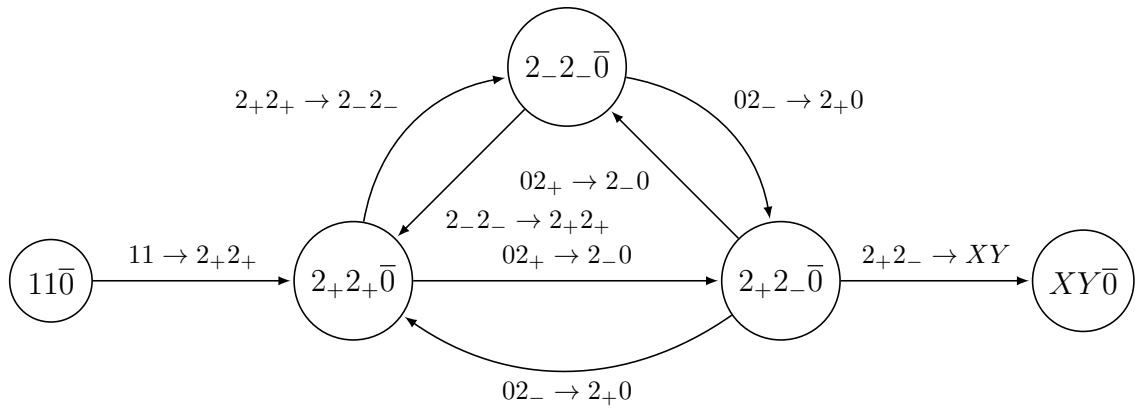

Figure 3.1: Graph of configurations for two occurrences of symbol σ in the input.

This protocol is Pavlovian because it corresponds to Example 2. Let us prove that it computes the threshold predicate $[x.\sigma \geq 3]$.

If there is no occurrence of input symbol σ in the input x, then the starting configuration is $(0\ldots0)$. In this case, no interaction allows agents to change their state. This argument can be also applied for the case where there is one occurrence of input symbol σ in the input x: the starting configuration is $(1)(0\ldots0)$.

Now we consider the case where there are two occurrences of input symbol σ in the input x. If the number of agents is 2, then, after the first interaction, the system switches between two configurations (2_+2_+) and (2_-2_-). Otherwise, i.e. if there are three or more agents in the population, then Figure 3.2.1 shows the graph of possible configurations (where $\overline{0}$ designates an arbitrary number of agents in state 0) and the possibility to switch from one configuration to another. All possible interactions in any of these configurations not shown in Figure 3.2.1 leave said configuration unchanged. Thus, in any reachable configuration from the initial $11\overline{0}$ configuration, all agents agree on output 0.

Finally, we consider the case where there are at least three occurrences of input symbol σ in the input x. First of all, the number of the agents being in state 0 never increases, so there are always, in any reachable configuration, at least three elements in $\{1, 2_+, 2_-, X, Y, \top\}$. Additionally, no agent can go to state 1 from any other given state. The principle of the proof of correctness is as follows: we will prove that from any configuration with at least three agents with states in $\{1, 2_+, 2_-, X, Y, \top\}$, there is a sequence of interaction increasing the number of agents in state $\top$ by at least one. This in turn means that, by iterating on such sequences, we can construct a sequence which leads us to a configuration were all agents are in state $\top$, which is a stable configuration in which everyone agrees on the correct output. The fairness hypothesis then ensures that any valid computation eventually leads to this configuration which will conclude the proof.

So, let us prove that from any configuration with at least three agents with states in $\{1, 2_+, 2_-, X, Y, \top\}$, there exists a series of interaction between those three agents which will increase the number of agents in state $\top$ by one. First let us consider the case where three agents are in states in $\{1, 2_+, 2_-, X, Y\}$. We will show that we can turn one of these agents

to state $\top$. If two of these agents are in state 1, the rule $11 \to 2_+2_+$ guarantees that we can have at least two agents with states states in $\{2_+, 2_-, X, Y\}$ which we will assume from now on.

If three agents are in states X or Y, then at least two are in the same state. Thus, by transition rule $XX \to \top\top$ or $YY \to \top\top$, a $\top$ can appear. If this is not the case, then at least one agent has a state s in $\{1, 2_+, 2_-\}$. If there is another agent in state $s' \in \{X, Y\}$, the interaction between s and s' creates a $\top$.

The final case is when all agents have states in $\{1, 2_+, 2_-\}$ (recall that at least two are in states 2_+ or 2_-). Then the remaining possible configurations, and their derivation,s are as follow:

- $\underline{1, 2_i}, 2_i \to \underline{2_{\bar{i}}2_i}2_i \to X\underline{Y2_i} \to X\top 2_-$ with $i \in \{+, -\}$
- $1\underline{2_+2_-} \to \underline{1X}Y \to \top XY$
- $\underline{2_i2_i}2_i \to 2_{\bar{i}}\underline{2_{\bar{i}}2_i} \to \underline{2_{\bar{i}}X}Y \to \top 2_iY$
- $2_i\underline{2_i2_{\bar{i}}} \to \underline{2_iX}Y \to \top 2_{i+1}Y$

In any case, we can bring one of the three agents to state $\top$. wThus if there are at least three agents in states $\{1, 2_+, 2_-, X, Y\}$, then we can increase the number of agents in state $\top$ by at least one.

If there are at most two agents in states $\{1, 2_+, 2_-, X, Y\}$, then there is necessarily at least one agent in state $\top$, because at least three agents are in non-0 states. If one of the agents in states $\{1, 2_+, 2_-, X, Y\}$ is actually in state 1 (resp. X and Y) then, by pairing it with the agent in state $\top$, it will be converted to $\top$. This again increases the number of agents in state $\top$ by one.

Otherwise, if exactly two agents are in states $\{2_+, 2_-\}$ and at least one agent is in state $\top$, then one of the following sequences of interactions will bring them all to state $\top$.

- $2_+\underline{2_+\top} \to 2_+\underline{2_+2_-} \to \underline{2_+Y}X \to \top\underline{2_-X} \to \top\top\top$
- $\underline{2_+2_-}\top \to X\underline{Y\top} \to \underline{X\top}\top \to \top\top\top$
- $2_-\underline{2_-\top} \to 2_-\underline{2_-2_+} \to \underline{2_-X}Y \to \top\underline{2_+Y} \to \top\top\top.$

If only one agent is in state 2_+ or 2_- and every other agent is either in state $\top$ or in state 0, then the transition rules $2_+\top \to 2_+2_-$ or $2_-\top \to 2_-2_+$ bring us back to the previous situation (and since both agents can then be converted to $\top$, the global number of agents in $\top$ will then indeed have been increased by one).

Finally, if no agents are in states $\{1, 2_+, 2_-, X, Y\}$, then every agent is in state 0 or $\top$ with at least three of them being in state $\top$. The rule $0\top \to \top\top$ then allows us to convert the 0s to $\top$.

Thus, from any configuration with at least three non-0 agents, the number of agents in state $\top$ can be strictly increased which, according to what we said earlier concludes our proof. $\square$

3.3 Some non-trivial Exclusive Pavlovian Protocols.

We will now show some other examples of what it is possible to compute using exclusive Pavlovian protocols. We will not however give a full logical characterisation of what it is possible to compute using such protocols. Indeed, unfortunately, such a characterisation still eludes us.

3.3.1 Leader Election

The classical solution [2] to the leader election problem [1] is the following:

$$\begin{cases} LL & \rightarrow & LN \\ LN & \rightarrow & LN \\ NL & \rightarrow & NL \\ NN & \rightarrow & NN \end{cases} \tag{3.2}$$

Where L is the state of a Leader. Notice that we use the terminology "leader election" as in [2] for this protocol, but that it may be considered more as a "mutual exclusion" protocol. Unfortunately, this protocol is non-symmetric, and hence not symmetric Pavlovian.

The problem here lies with the first rule, since one wants two leaders to become only one. If the two leaders are identical, this is clearly problematic with symmetric rules. However, the leader election problem can actually be solved by an Exclusive Pavlovian protocol, at the cost of additional complexity in the protocol.

Theorem 6 *The following exclusive Pavlovian protocol solves the leader election problem, in any population of size $n \geq 3$.*

$$\begin{cases} L_1L_2 & \rightarrow & L_1N \\ L_1N & \rightarrow & NL_2 \\ L_2N & \rightarrow & NL_1 \\ NN & \rightarrow & NN \\ L_2L_1 & \rightarrow & NL_1 \\ NL_1 & \rightarrow & L_2N \\ NL_2 & \rightarrow & L_1N \\ L_1L_1 & \rightarrow & L_2L_2 \\ L_2L_2 & \rightarrow & L_1L_1 \end{cases} \tag{3.3}$$

Proof. Starting from a configuration containing at least one L, all configurations will eventually have exactly one leader, that is one agent in state L_1 or L_2. Indeed, the first rule and the fifth rule strictly decrease the number of leaders whenever there are more than two leaders. The other rules preserve the number of leaders, and are made such that an L_1 can always be transformed into an L_2 and vice-versa. Hence, they are made such that a configuration where first or fifth rule applies can always be reached whenever there are more than two

[1] starting from a configuration with ≥ 1 leaders, eventually exactly one leader survives

leaders. The fact that it solves the leader election problem then follows from the hypothesis of fairness in the definition of computations.

This is an Exclusive Pavlovian protocol, since it corresponds to the following pay-off matrix.

		Opponent		
		L_1	L_2	N
Player	L_1	-3	0	-3
	L_2	-1	-3	-3
	N	-2	-3	0

□

Note that this is a rare occurrence where using the general method to produce a symmetric population protocol computing the same predicate as a given (non-symmetric) protocol yields one which also happens to be an exclusive Pavlovian protocol. Exclusivity is key here since we require that $BR_{\neq L_1}(L_1) = L_2 \neq L_1 = BR_{\neq L_2}(L_1)$.

3.3.2 Majority

We call majority problem the computation of the the predicate $\psi(x) = [x.\sigma \geq x.\sigma']$, where σ and σ' are the only two input symbols, and $x.\sigma$ is the number of occurrences of input symbol σ in input x.

Proposition 2 *The majority problem can be solved by an exclusive Pavlovian population protocol.*

Proof. We claim that the following protocol outputs 1 if there are more σ than σ' in the initial configuration and 0 otherwise. The transitions are

$$\begin{cases} NY & \rightarrow & YY \\ YN & \rightarrow & YY \\ N\sigma & \rightarrow & Y\sigma \\ \sigma N & \rightarrow & \sigma Y \\ Y\sigma' & \rightarrow & N\sigma' \\ \sigma' Y & \rightarrow & \sigma' N \\ \sigma\sigma' & \rightarrow & NY \\ \sigma'\sigma & \rightarrow & YN \end{cases} \tag{3.4}$$

with

- $\Sigma = \{\sigma, \sigma'\}$
- $Q = \{\sigma, \sigma', Y, N\}$,

- $\omega(\sigma) = \omega(Y) = 1$,
- $\omega(\sigma') = \omega(N) = 0$.

In this protocol, the states Y and N are "neutral" elements for our predicate but they should be understood as *Yes* and *No*. They are the "answers" to the question: are there more σs than σ's. This protocol is made such that the numbers of σ and σ' are preserved except when a σ meets a σ'. In that latter case, the two agents are deleted and transformed into a Y and a N.

If there are initially strictly more σ than σ', from the fairness condition, each σ' will be paired with a σ and at some point no σ' will left. By fairness and since there is still at least a σ, a configuration containing only σ and Ys will be reached. Since in such a configuration, no rule can modify the state of any agent, and since the output is defined and equals to 1 in such a configuration, the protocol is correct in this case

By symmetry, one can show that the protocol outputs 0 if there are initially strictly more σ' than σ.

Suppose now that, initially, there are exactly the same number of σ and σ'. By fairness, there exists a step when no more agents in the state σ or σ' left. Note that at the moment where the last σ is matched with the last σ', a Y is created. Since this Y can be "broadcasted" over the Ns, in the final configuration all agents are in the state Y and thus the output is correct.

This protocol is Pavlovian, since it corresponds to the following pay-off matrix.

		Opponent			
		N	Y	σ	σ'
	N	1	-1	-1	1
Player	Y	0	1	1	-1
	σ	0	0	0	-1
	σ'	0	0	-1	0

□

3.3.3 Counting up to 2^k

Theorem 7 *For any $k \geq 2$, there is an exclusive Pavlovian protocol that computes the threshold predicate $[x.\sigma \geq 2^k]$, which is true when there are at least 2^k occurrences of input symbol σ in the input x.*

To prove Theorem 7, we will show that the following protocol $\mathcal{P}$ counts up to 2^k.

- $Q = \{0, \top\} \bigcup_{i=1}^{2^{k-1}-1} \{i_+, i_-\}$.
- $\Sigma = \{0, \sigma\}$.

- $\iota(\sigma) = 1_+$, $\iota(0) = 0$.
- $\omega = 1_{\{\top\}}$.
- Its transition function can be written as follows:

For any $n \in [1, 2^k)$

$$00 \to 00$$
$$0n_+ \to n_-0$$
$$0n_- \to n_+0$$
$$0\top \to \top\top$$

If $n \neq m$,

$$n_-m_+ \to m_-n_+$$
$$n_+m_- \to m_+n_-$$
$$n_-m_- \to m_+n_+$$
$$n_+m_+ \to m_-n_-$$

For any $n \in [1, 2^k)$

$$n_+n_+ \to n_-n_-$$
$$n_-n_- \to n_+n_+$$
$$n_-n_+ \to (2n)_+(2n+1)_+ \text{ if } n < 2^k$$
$$n_-n_+ \to \top\top \text{ if } 2^{k-1} \leq n < 2^k$$
$$n_-\top \to n_-n_+$$
$$n_+\top \to n_+n_-$$
$$\top\top \to \top\top$$

This system is exclusive Pavlovian because we can define Stable(q), $max(q)$ and $\delta_q(max(q))$ for any state q as follows:

q	Stable(q)	$max(q)$	$\delta_q(max(q))$
0	$\{0, \top\}$	0	$-$
$\top$	$Q \setminus \{0\}$	$\top$	$-$
n_+	$\emptyset$	n_-	$(2n+1)_+$
n_-	$\emptyset$	n_+	$(2n)_+$

To simplify the proof, we will introduce another protocol which is very similar to $\mathcal{P}$ and indeed computes the same predicate. We will then show that this new predicates computes the predicate $[x.\sigma \geq 2^k]$, thereby proving that $\mathcal{P}$ does indeed count up to 2^k.

Let us consider the protocol $\mathcal{P}'$ which differs from $\mathcal{P}$ only in its transition function given by:

For any $n \in [1, 2^k)$

$$00 \to 00$$
$$0n_+ \to 0n_-$$
$$0n_- \to 0n_+$$
$$0\top \to \top\top$$

If $n \neq m$,

$$\begin{array}{c} n_-m_+ \rightarrow n_+m_- \\ n_+m_- \rightarrow n_-m_+ \\ n_-m_- \rightarrow n_+m_+ \\ n_+m_+ \rightarrow n_-m_- \end{array}$$

For any $n \in [1, 2^k)$

$$\begin{array}{l} n_+n_+ \rightarrow n_-n_- \\ n_-n_- \rightarrow n_+n_+ \\ n_-n_+ \rightarrow (2n)_+(2n+1)_+ \text{ if } n < 2^k \\ n_-n_+ \rightarrow \top\top \text{ if } 2^{k-1} \leq n < 2^k \\ \ n_-\top \rightarrow n_-n_+ \\ \ n_+\top \rightarrow n_+n_- \end{array}$$

$$\top\top \rightarrow \top\top$$

The transition function of $\mathcal{P}'$ is the transition function of $\mathcal{P}$ in which some transition rules of the form $ab \rightarrow a'b'$ have been replaced by corresponding (and computationally equivalent) rule $ab \rightarrow b'a'$. The anonymity of agents in a Population Protocol implies that from a Population Protocol point of view, protocols $\mathcal{P}$ and $\mathcal{P}'$ are equivalent and compute the exact same predicate (if any). The difference is that $\mathcal{P}'$ is not Pavlovian. However, proving that $\mathcal{P}'$ computes the desired predicate will yield that the same holds for $\mathcal{P}$. We will now study $\mathcal{P}'$ instead of $\mathcal{P}$ in order to simplify the proof.

Let a be an arbitrary agent of the population. Let q and C be respectively a state in Q and a configuration.

Let us define $C(a)$ the state of agent a in configuration C and $C^{\#}(q)$ the number of agents in state q in configuration C.

Finally, we define $v(q)$ the level of a state q to be the real number defined for $n \in \{1, \ldots, 2^k\}$ by $v(0) = 0, v(n_+) = v(n_-) = n, v(\top) = 2^k$.

Lemma 6 *For any two configurations C and C' such that $C \rightarrow C'$ in protocol P', we have either $v(C'(a)) \geq v(C(a))$, or $C(a) = \top$.*

Proof. This means that an agent that is not in state $\top$ cannot be sent back in the execution of states which is easily verified by looking at the transition rules for $\mathcal{P}'$. Note that this key property does not hold for $\mathcal{P}$ and will allow us to simplify the following proofs. □

Lemma 7 *Let $C_0, ..., C_{i_1}$ be an execution of configurations such that for every i such that $0 \leq i < i_1, C_i \rightarrow C_{i+1}$ and $C_0^{\#}(\top) = ... = C_{i_1}^{\#}(\top) = 0$ and for every $q \notin \{0, 1_+\}, C_0^{\#}(q) = 0$.*

Then for every n and every i such that $1 \leq n \leq 2^{k-1}$ and $i \leq i_1$ it holds that

$$C_i^{\#}(n_+) + C_i^{\#}(n_-) \leq C_0^{\#}(1_+)2^{-\lfloor log(n) \rfloor}.$$

Proof. We will actually prove a slightly stronger statement by induction over n :

In the execution of configurations $C_0, \ldots, C_{i_1}$ no more than $C_0^{\#}(1_+)2^{-\lfloor log(n) \rfloor}$ agents may ever reach states n_+ or n_-.

Calling $S(n), n \geq 1$ the set of agents that ever reach states n_+ or n_- in this computation, we will prove that $\text{Card}(S(n)) \leq C_0^{\#}(1_+)2^{-\lfloor log(n) \rfloor}$.

First, we note that this is an execution of configurations in which no agent ever reaches state $\top$. Because of this and the previous lemma, an agent can only have growing value in the execution $C_0, ..., C_{i_1}$. In addition, agents initially in state 0 can never change their state (this would only be possible if they encountered an agent in state $\top$). Thus, the set of agents with state different from 0 is the same in every configuration in the execution. It follows naturally that no more than $C_0^{\#}(1_+)$ agents may ever reach states 1_+ or 1_-.

To prove the statement for a given $n \geq 2$ assume by induction that it is true for all $k < n$. . It follows from the transition rules that $S(k) \subset S(\lfloor \frac{k}{2} \rfloor)$ by looking at the state an agent was before it ever reached level k. This also holds for level n. In fact any given agent in $S(n)$ first reaches level n through an interaction of type $\frac{n}{2}_+ \frac{n}{2}_- \rightarrow n_+(n+1)_+$ if n is even (and $\frac{n-1}{2}_+ \frac{n-1}{2}_- \rightarrow (n-1)_+ n_+$ if n is odd). Thus, it appears that at most half the agents in $S(\lfloor \frac{n}{2} \rfloor)$ ever reach level n (the other half either being sent to level $n-1$ or $n+1$ depending on the parity of n or never going beyond level $\lfloor \frac{n}{2} \rfloor$).

Therefore $Card(S(n)) \leq \frac{Card(S(\frac{n}{2}))}{2} \leq C_0^{\#}(1_+)2^{-\lfloor log(n) \rfloor}$.

□

Lemma 8 *Let $C_0, C_1, \ldots, C_i, \ldots$ be a correct execution of protocol P'. If $C_0^{\#}(1_+) < 2^k$ then $\forall i \geq 0, C_i^{\#}(\top) = 0$.*

Proof. Contrariwise, let us assume that there exists at least one configuration C_i with at least one agent in state $\top$. Let us consider C_{i_0} the earliest such configuration. This means that the transition $C_{i_0-1} \rightarrow C_{i_0}$ happens through an encounter of two agents $n_+ n_-$ with $n \in [2^{k-1} ... 2^k - 1]$.

Then $C_0, \ldots, C_{i_0-1}$ is a non-empty execution fitting the conditions in the previous lemma. Which tells us that $C_{i_0-1}^{\#}(n_+) + C_{i_0-1}^{\#}(n_-) \leq 2^{-\lfloor log(n) \rfloor} C_0^{\#}(1_+)$. Since $C_{i_0-1}^{\#}(n_+) + C_{i_0-1} \geq 2$ for this interaction to be possible, we have:

$C_0^{\#}(1_+) \geq 2^{\lfloor log(n) \rfloor} \geq 2^k$. □

We have now proved that if strictly less than 2^k agents are in state 1_+ initially, no agent will ever reach state $\top$ and thus all agents will always agree on output 0 and the computation will be correct. We will now prove that the computation will be correct if at least 2^k agents are initially in state 1_+.

Lemma 9 *For any configuration C in which at least 2^k agents are in non-zero states, there exists a configuration C' such that $C \rightarrow^* C'$ and $C'^{\#}(\top) \geq 1$.*

Proof. If $C^{\#}(\top) \geq 1$, just take $C' = C$. Now assume $C^{\#}(\top) = 0$, then the pigeonhole principle insures that there are at least 2 agents at the same level. Let k be the smallest

level with at least 2 agents. We will now prove that there is a finite sequence of configurations forming valid computation steps that increases the value of k. We shall now differentiate the cases where $C^{\#}(k_+) + C^{\#}(k_-) > 2$ and $C^{\#}(k_+) + C^{\#}(k_-) = 2$.

If $C^{\#}(k_+) + C^{\#}(k_-) > 2$, if at least two agents in level k are in different states, we can have those two interact according to rule $k_+k_- \to 2k_+(2k+1)_-$ (or rule $k_+k_- \to \top\top$) to diminish the number of agents in level k by two (and not create agents in lower levels) or, if they are all in the same state, uniformity can be broken by having two of them interact via rules $k_+k_+ \to k_-k_-$ or $k_-k_- \to k_+k_+$ to create two agents in the other level-k state and be brought back to the previous configuration. By iterating on those two actions, we can bring the number of agents at level k to two or less. If only one remains, we have achieved our goal, otherwise, we handle the two remaining agents as follows: If $C^{\#}(k_+) + C^{\#}(k_-) = 2$, we again, differentiate: if $C^{\#}(k_+) = C^{\#}(k_-) = 1$ then selecting the two agents in states k_+ and k_- for an interaction $k_+k_- \to 2k_+(2k+1)_-$ will yield the desired result. Otherwise, if both agents at level k are in the same state, then we can break this symmetry by having one interact with any other non-zero agent to come back to the previous case. Such a non-zero and non-level-k agent exists since there are at least 2^k non zero agents.

Thus, from configuration C we have an execution $C \to^* C_1$ where $C_1^{\#}(k_-) < 2$. If $C_1^{\#}(\top) > 0$ we have achieved our desired result, otherwise, we can reiterate on C_1, knowing that the minimal level with at least two agents in C_1 is $k_1 > k$ by construction. Iterating this process gives us an execution $C \to^* C_1 \to^* ... \to^* C_j$ such that either an agent in state $\top$ appears or we have a corresponding execution $k < k_1 < ... < k_j$ of minimal level with at least two agents. Since this strictly growing execution is upper bounded by 2^k it is finite which guarantees that after a certain amount of iterations, at least one agent will reach state $\top$. □

For any configuration C with at least one agent in a non-zero state, we denote by $m(C)$ the smallest level with non-zero count in C. Note that $m(C)$ can never decrease: no interaction rule creates agents in a level lower than those already existing in the system.

Lemma 10 *For any configuration C such that $C^{\#}(\top) \geq 1$, and $m(C) < 2^k$, there exists a configuration C' such that $C \to^* C'$ and $m(C') > m(C)$.*

Proof. Similarly to what was done before, if there are at least two agents at level $m(C)$ then we can reduce the number of agents at level $m(C)$ by two and, iterating the process bring it to at most 1. Note that this process can be done by preserving the existence of agents in state $\top$. If we are left with no agents in state $m(C)$, we have achieved our goal. If not, we are left with a single agent a in state $m(C)$ and all other agents in states greater than $m(C)$, including at least one agent a' in state $\top$. Assume that a is in state k_+ (with, $k = m(C)$), the case k_- being symmetric). Then we can eliminate our final agent a through two interactions with a' :

$$k_+\top \to k_+k_- \to 2k_+(2k+1)_+.$$

This brings us to a configuration C' such that $C'^{\#}(m(C)) = 0$ and thus, $m(C') > m(C)$. □

Lemma 11 *From any configuration C in which at least 2^k agents are in non-zero states, there exists a computation leading to a configuration in which all agents are in state $\top$.*

Proof. This is achieved mainly by iteration of the previous two lemmas: from configuration C, following Lemma 9, reach a configuration C' where there is at least one agent in state $\top$. If some non-zero agents are not in state $\top$, increase the minimal non-zero level in the system per Lemma 10. Iterate until the minimal non-zero level is 2^k, ie. all agents are either in state $\top$ or in state 0. Then use the transition rule $0\top \to \top\top$ to convert all remaining agents from state 0 to state $\top$. Note that such a configuration is trivially stable. □

This now allows us to prove Theorem 7.

Proof of Theorem 7 From Lemma 11, the fairness property ensures that any fair computation of protocol $\mathcal{P}'$ starting in a configuration with at least 2^k agents in state 1_+ stably converges to a configuration in which all agents are in state $\top$ and thus agree on output 1. Contrariwise, if the initial configuration holds strictly less than 2^k agents in state 1_1 then Lemma 8 guaranties that all agents will always agree on output 0.

Thus protocol $\mathcal{P}'$ computes the predicate $[x.\sigma \geq 2^k]$ and, since they are equivalent, so does $\mathcal{P}$. □

3.4 Conclusion

In this chapter, we showed that restricting population protocols to those with symmetric rules did not, in fact, restrict their computational power because any population protocol could be simulated by a symmetric population protocol if there are at least three agents in the population. The cost of such a simulation is a doubling of the set of states available to the agents. We showed that the same result does not, however, hold for Pavlovian protocols. Indeed, symmetric Pavlovian protocols seem to be quite limited in power as they are unable even to detect if 3 or more occurrences of a single input symbol can be found in the population or not.

We managed to circumvent this limitation of symmetric Pavlovian protocols by considering exclusive Pavlovian protocols in which an agent who is dissatisfied with his gain in a given encounter is forced to change his state even if the current state is already the best possible response. We proved that such exclusive Pavlovian protocols were strictly more powerful than symmetric Pavlovian protocols as they can count up to 3 or to 2^k for any $k \geq 2$.

The structure of the protocol counting up to 2^k is designed to take advantage of the pairwise nature of the interaction between agents and uses it to spread the agents across the possible states while trying to avoid duplication. This is why it works for thresholds of form $x.\sigma \geq 2^k$ but is not necessarily adaptable to the more general case of threshold predicates, especially for weighted thresholds. Indeed, recall that threshold predicates in classical population protocols are computed by protocols in which the total sum of the values stored in the system is conserved throughout computation whereas here, the agents

seem to try and store the number of agents that have been met and use the spread across distinct values modulo 2^i to avoid counting the same agents multiple times.

It seems highly unlikely, therefore, that a similar construction would work for a different type of threshold. That is not to say that another construction is impossible but we do not know of any such protocol as of now. This means that we do not know the exact computational power of exclusive Pavlovian protocols, which is still open for future work.

Part II

Computing with Large Population Protocols

A key aspect of population protocols [2], from which only a few variants have deviated is robustness to size: a given protocol (and the size of available memory it requires from the individual agents) should be fixed and independent on the size of any given population running it. This in particular implies some form of anonymity of the agents since, with a finite memory of predetermined fixed size, agents could not possibly hope to remember enough information to even store individual unique identifiers in an arbitrarily large population.

This robustness to size naturally lead us to try and understand the behaviour of population protocols when the size of the population is "extremely large", calling them *Large-Population Protocols*, or ***LPP***s for short. Indeed, many traditional models capturing the dynamics of populations have assumed populations to be extremely large, often approximating them with a virtual infinite population. Such population models include *Predator-Prey* systems, the *Lotka-Volterra* dynamics and the *replicator* dynamics and, generally, Evolutionary Games Theory [49]. The *Lotka-Volterra* dynamics are known to be equivalent to *replicator* dynamics for example (see [34]) and have been shown to be captured by a subclass of dynamics corresponding to a subclass of population protocols [21].

Given a population of size n, a population protocol traditionally considers the vector $X(t) = (X_1(t), ..., X_{|Q|}(t)) \in \mathbb{N}^Q$ where $X_i(t)$ is the number of agents in state i at time t, whereas the models for the dynamics of population referred to above consider the fraction vector $x(t) = \frac{X(t)}{n}$. Indeed, when the total number n of agents in the population becomes huge, the evolution of a single agent has little impact of the overall structure of the population. Moreover, when the population becomes infinite, the exact count of agents in a given state becomes meaningless and should be replaced by the fraction. Additionaly, the adversary scheduler considered is no longer simply assumed to be fair but to pick the interacting pair uniformly at random in the population.

Population dynamics stemming from Game Theory model the behaviour of an inifinite population of agents repeatedly playing a game and updating their current strategy (or state) depending on the result of their individual score compared to population-wide results or predetermined thresholds. This similarity with our study of Pavolvian protocols, though not central in our motivation to study LPPs, also makes LPPs a natural extention of Pavlovian protocols.

This part the thesis is divided into two chapters. The first chapter introduces our new model of *Large Population Protocols* or *LPP*s. After showing how the behaviour of such a protocol can be approximated by the solution to an ordinary differential equation when the population grows to infinity, we will use this to define a notion of computation for LPPs.A LPP is said to compute a real number $\nu \in [0, 1]$ if it converges to an equilibrium in which the fraction of agents are in a subset of states called *marked* states. The second chapter focuses on characterising the set of numbers computable by such a LPP an proves this to be exactly the set of algebraic numbers in $[0, 1]$.

Chapter 4

Large Population Protocols

In this chapter, we define a new model, derived from population protocols. This model aims at capturing the behaviour of very large populations of anonymous agents with a fixed size memory interacting in pairs. We call this model *Large-Population Protocols* from its similarity to population protocols [2]. We then analyse the behaviour of such protocols, first illustrating this analysis on a toy example. This will allow us to prove that the behaviour of an LPP can be correctly approximated by the solution to an ordinary differential equation (ODE). This leads us to define a notion of computation for LPPs. A LPP is said to compute a real number ν if it converges to an equilibrium of the corresponding ODE in which a fraction ν of agents are in a subset of states called *marked* states. Because of the approximations involved and the stochastic nature of the actual dynamic of LPPs, this convergence involves both the size of the population (considered going to infinity) as well as the execution time of the protocol (going to infinity too). We show that, to approach such a computed number ν within a margin of ϵ with probability at least $\mu \in]0, 1[$ can be guaranteed with a population of size polynomial in $\frac{1}{\epsilon}$ after a time polynomial in $\frac{1}{\epsilon}$. The results on the toy example were published in [15]. Other parts of this chapter are to appear in [16].

In keeping with classical population protocols, we consider our population of n anonymous agents, each of which can be in finitely many possible states, from a finite set $Q = \{q_1, \ldots, q_m\}$. This population is evolving in a synchronous discrete-time system. and, 1t each round, two agents a and b are selected uniformly at random from among the n agents. These agents then interact according to a set of transition rules Δ of of the form $q_i\ q_j \rightarrow q_k\ q_l$ which we shall also denote functionally as $(q_k, q_l) = \Delta(q_i, q_j)$.

We will, however, not consider the population from the point of view of counts of agents in any given state. Instead, we consider the fraction of the population that currently is in any given state. Another departure from traditional population protocols is that the selection of the pair of agents interacting is performed uniformly at random, independently from the past (instead of just assuming a fairness hypothesis restricting the power of the adversary controlling the selection of agents interacting). Indeed, when the size of the population goes to infinity, uniform sampling of agents appears to be a natural way to extend the fairness hypothesis used in classical population protocols while being more practical to study. Moreover, uniform sampling is consistent with the interpretation of agents as autonomous

entities moving at random (see, e.g. [7] for a discussion on random adversaries in finite state systems).

The final divergence from population protocols in *LPPs* is in the notion of computation that we will formally define later. Indeed, the notion of *stable* computation used by [2] does not consider the fractions of agents in the different states. Intuitively, we will be computing a real number $\nu \in [0, 1]$ by designing protocols ensuring that, when the population is large enough, the fraction of agents in a specific subset of states stably converges to the desired number ν under some proximity conditions on the initial configuration. The main result of this chapter, after the formal definition of this notion of computation is the exact characterization of the real numbers that can be computed by an *LPP* in this sense. They are all algebraic numbers of $[0, 1]$.

4.1 An illustrative example

To illustrate the analysis of our new computing model, and its differences from classical population protocols, we will consider the example population protocol with set of possible states $Q = \{+, -\}$, and the following transition relation:

$$\begin{cases} ++ & \rightarrow & +- \\ +- & \rightarrow & ++ \\ -+ & \rightarrow & ++ \\ -- & \rightarrow & +- \end{cases} \tag{4.1}$$

Using the classical definition of computation in a Population Protocol, this specific protocol does not stably compute anything. Indeed, if we put aside the special configuration where all agents are in state $-$ (which is immediately left in any next round), any configuration is reachable from any configuration in a finite number of steps. This combined with the fairness property of stable computations, ensures that any configuration will be reached infinitely often. We will however show that things are different when considering the proportion $p(k)$ of agents in state $+$ at time step k.

As mentioned earlier, we suppose that at each time step, two distinct agents are picked uniformly at random among the n agents. Keeping with the anonymity of the agents, the whole system ins described by the number $n_+(k)$ of agents in state $+$ at step k, from which we can deduce the proportion $p(k) = \frac{n_+(k)}{n}$.

We are then reduced to determine the evolution of the Markov chain

$$(p(k))_{k\in\mathbb{N}} \in \left\{\frac{0}{n}, \frac{1}{n}, \ldots, \frac{n}{n}\right\}.$$

Interestingly enough, the same reasons that prevented the protocol from (stably) computing anything in the classical sense ensure that $(p(k))$ is an irreducible Markov chain in $\{\frac{1}{n}, \ldots, \frac{n}{n}\}$. Let us now compute the transition probabilities of this irreducible Markov chain. We have

$$p(k + 1) - p(k) \in \{-1, 1\}.$$

Then, we have to determine for each $i = 1, 2, \ldots, n$

$$\pi^{(n)}(\tfrac{i}{n} \to \tfrac{i-1}{n}) := \mathbb{P}\left(p(k+1) = \frac{i-1}{n} \mid p(k) = \frac{i}{n}\right),$$
$$\pi^{(n)}(\tfrac{i}{n} \to \tfrac{i+1}{n}) := \mathbb{P}\left(p(k+1) = \frac{i+1}{n} \mid p(k) = \frac{i}{n}\right).$$

Assume that $p(k) = i/n$. Then $p(k)$ decreases only if the two agents sampled are in state $+$. That is,

$$\pi^{(n)}(\tfrac{i}{n} \to \tfrac{i-1}{n}) = \frac{\binom{i}{2}}{\binom{n}{2}} = \frac{i(i-1)}{n(n-1)}.$$

In any other case, $p(k)$ increases by 1 :

$$\begin{aligned}\pi^{(n)}(\tfrac{i}{n} \to \tfrac{i+1}{n}) &= 1 - \pi^{(n)}(\tfrac{i}{n} \to \tfrac{i-1}{n}) \\ &= 1 - \frac{i(i-1)}{n(n-1)}.\end{aligned}$$

A consequence of the ergodic theorem is that the chain $(p(k))$ admits a unique stationary distribution μ. By definition, this unique stationary distribution is the only application

$$\mu : \left\{\frac{1}{n}, \ldots, \frac{n}{n}\right\} \to [0, 1]$$

such that

1. $\sum_{i=1}^{n} \mu(i/n) = 1$.
2. μ satisfies the *balance equation*, *i.e.* for each i

$$\mu(\frac{i}{n}) = \mu(\frac{i-1}{n})\pi^{(n)}(\tfrac{i-1}{n} \to \tfrac{i}{n}) + \mu(\frac{i+1}{n})\pi^{(n)}(\tfrac{i+1}{n} \to \tfrac{i}{n}).$$

Without considering the exact expression of μ, we can guarantee that, as the unique solution to a rational system, μ is an element of $\mathbb{Q}^n$. Hence, its mean $\sum_i \mu(i/n)i/n$ is also a rational number, which we denote by $p^{(n)}$.

A second consequence of the ergodic theorem is that, almost surely,

$$\frac{p(1) + p(2) + \ldots + p(k)}{k} \overset{k\to\infty}{\to} p^{(n)}.$$

However, we will prove that when n goes to infinity, the mean value of $p(k)$ converges to the irrational number $\sqrt{2}/2$.

At first, let us give an intuition of this result by some informal calculations. Observe that

$$\pi^{(n)}(\tfrac{i}{n} \to \tfrac{i-1}{n}) = p^2(k)\frac{n}{n-1} - p(k)\frac{1}{n-1},$$

and write

$$\begin{aligned}\mathbb{E}[n_+(k+1) - n_+(k) \mid n_+(k)] &= \pi^{(n)}(\tfrac{i}{n} \to \tfrac{i+1}{n}) - \pi^{(n)}(\tfrac{i}{n} \to \tfrac{i-1}{n}) \\ &= 1 - 2\pi^{(n)}(\tfrac{i}{n} \to \tfrac{i-1}{n}) \\ &= 1 - 2p^2(k)\tfrac{n}{n-1} + p(k)\tfrac{2}{n-1}\end{aligned} \tag{4.2}$$

Assymptoticaly, when n becomes very large, the right-hand term is close to $1 - 2p(k)^2$. Now, when k goes to infinity, if there is some convergence of the mean proportion of $+$, the average variation of $p(k)$ must vanish to 0. Thus, the limit of $1 - 2p(k)^2$ must be close to 0 and the limit of $p(k)$ must be approximately $\frac{\sqrt{2}}{2}$.

We will now prove this convergence formally.

4.1.1 A General Theorem about Approximation of Diffusions

Let us introduce a theorem that we will use to prove our result and, later on, to analyse the general model. The theorem comes from [48] but we will use the formulation of it given in [22] (Theorem 5.8 page 96).

Suppose that for all integers $n \geq 1$, we have an homogeneous Markov chain $(Y_k^{(n)})$ in $\mathbb{R}^d$ with transition kernel $\pi^{(n)}(x, dy)$, meaning that the law of $Y_{k+1}^{(n)}$, conditioned on $Y_0^{(n)}, \cdots, Y_k^{(n)}$, depends only on $Y_k^{(n)}$ and is given, for every Borelian B, by

$$P(Y_{k+1}^{(n)} \in B | Y_k^{(n)}) = \pi^{(n)}(Y_k^{(n)}, B),$$

almost surely.

Define for $x \in \mathbb{R}^d$,

$$\begin{aligned}b^{(n)}(x) &= n \int (y-x)\pi^{(n)}(x, dy), \\ a^{(n)}(x) &= n \int (y-x)(y-x)^* \pi^{(n)}(x, dy), \\ K^{(n)}(x) &= n \int (y-x)^3 \pi^{(n)}(x, dy), \\ \Delta_\epsilon^{(n)}(x) &= n\pi^{(n)}(x, B(x,\epsilon)^c),\end{aligned}$$

where $B(x, \epsilon)^c$ denotes the complement of the ball with radius ϵ, centered at x. In other words,

$$b^{(n)}(x) = n\mathbb{E}_x[(Y_1 - x)],$$

and

$$a^{(n)}(x) = n\mathbb{E}_x[(Y_1 - x)(Y_1 - x)^*]$$

where $\mathbb{E}_x$ stands for "expectation starting from x", that is,

$$\mathbb{E}_x[(Y_1 - x)] = \mathbb{E}[(Y_1 - x)|Y_0 = x].$$

The coefficients $b^{(n)}$ and $a^{(n)}$ can be interpreted as the instantaneous drift and the variance (or matrix of covariance) of $X^{(n)}$.

Define

$$X^{(n)}(t) = Y^{(n)}_{\lfloor nt \rfloor} + (nt - \lfloor nt \rfloor)(Y^{(n)}_{\lfloor nt+1 \rfloor} - Y^{(n)}_{\lfloor nt \rfloor}).$$

Theorem 8 (Theorem 5.8, page 96 of [22]) *Suppose that there exist some continuous functions a, b, such that for all $R < +\infty$,*

$$\lim_{n\to\infty} sup_{|x|\leq R}|a^{(n)}(x) - a(x)| = 0$$
$$\lim_{n\to\infty} sup_{|x|\leq R}|b^{(n)}(x) - b(x)| = 0$$
$$\lim_{n\to\infty} sup_{|x|\leq R}\Delta^{(n)}_{\epsilon} = 0, \forall \epsilon > 0$$
$$\sup_{|x|\leq R} K^{(n)}(x) < \infty.$$

With σ a matrix such that $\sigma(x)\sigma^*(x) = a(x)$, $x \in \mathbb{R}^d$, we suppose that the stochastic differential equation

$$dX(t) = b(X(t))dt + \sigma(X(t))dB(t), \qquad X(0) = x, \tag{4.3}$$

has a unique weak solution for all x. (This is in particular the case if it admits a unique strong solution.)

Then for all sequences of initial conditions $Y^{(n)}_0 \to x$, the sequence of random processes $X^{(n)}$ converges in law *to the diffusion given by* (4.3)*. In other words, for all function $F : \mathcal{C}(\mathbb{R}^+, \mathbb{R}) \to \mathbb{R}$ bounded and continuous, one has*

$$\lim_{n\to\infty} \mathbb{E}[F(X^{(n)})] = \mathbb{E}[F(X)].$$

4.1.2 Proving convergence of our example.

Returning to our toy example from Section 4.1, we will now use Theorem 8 to prove convergence of $p(k)$ to $\frac{\sqrt{2}}{2}$ when both n and k grow to infinity.

Consider $Y^{(n)}_i$ as the homogeneous Markov chain corresponding to $p(k)$, when n is fixed. From the previous discussions, $\pi^{(n)}(x, .)$ is a weighted sum of two Dirac distributions that weight $x - \frac{1}{n}$ and $x + \frac{1}{n}$, with respective probabilities π_{-1} and π_{+1}, whenever x is of type $\frac{i}{n}$ for some i.

From Equation (4.2) we have

$$\mathbb{E}[p(k+1) - p(k)|p(k)] = \frac{1}{n}(1 - 2p(k)^2\frac{n}{n-1} + p(k)\frac{2}{n-1}), \tag{4.4}$$

which yields

$$b^{(n)}(x) = 1 - 2p(k)^2\frac{n}{n-1} + p(k)\frac{2}{n-1},$$

when $x = i/n$. Now, $(p(k+1) - p(k))^2 = \frac{1}{n^2}$, and hence

$$a^{(n)}(x) = \frac{1}{n},$$

when $x = i/n$. Taking $a(x) = 0$ and $b(x) = 1 - 2x^2$, we get

$$\lim_{n\to\infty} sup_{|x|\leq R}|a^{(n)}(x) - a(x)| = 0$$

$$\lim_{n\to\infty} sup_{|x|\leq R}|b^{(n)}(x) - b(x)| = 0$$

for all $R < +\infty$. Since the jumps of $Y^{(n)}$ are bounded in absolute value by $\frac{1}{n}$, $\Delta_\epsilon^{(n)}$ is null, as soon as $\frac{1}{n}$ is smaller than ϵ, and so

$$\lim_{n\to\infty} sup_{|x|\leq R}\Delta_\epsilon^{(n)} = 0, \forall \epsilon > 0.$$

Finally, $K^{(n)}(x)$ is finitely bounded over any ball of radius R and

$$\sup_{|x|\leq R} K^{(n)}(x) < \infty.$$

Now, (ordinary and deterministic) differential equation

$$dX(t) = (1 - 2X^2)dt \tag{4.5}$$

has a unique solution for any initial condition.

It follows from Theorem 8 that the sequence of random processes $X^{(n)}$ defined by

$$X^{(n)}(t) = Y^{(n)}_{\lfloor nt \rfloor} + (nt - \lfloor nt \rfloor)(Y^{(n)}_{\lfloor nt+1 \rfloor} - Y^{(n)}_{\lfloor nt \rfloor})$$

converges in law to the unique solution of differential equation (4.5).

Any solution of ordinary differential equation (4.5) converges to $\frac{\sqrt{2}}{2}$. Indeed, doing the change of variable $Z(t) = X(t) - \frac{\sqrt{2}}{2}$, we get

$$dZ(t) = -2(Z + \sqrt{2})Zdt, \tag{4.6}$$

that converges to 0.

Coming back to $p(k)$ using the definition of $X^{(n)}(t)$, we hence get the following result.

Theorem 9 *For all t,*

$$p(\lfloor nt \rfloor) = \frac{\sqrt{2}}{2} + Z_n(t),$$

where $Z_n(t)$ converges in law when n goes to infinity to the (deterministic) solution of the ordinary differential equation (4.6)*. Solutions of this ordinary differential equation go to* 0 *at infinity.*

Theorem 9 implies that $p(k)$ must converge to $\frac{\sqrt{2}}{2}$ when k and n go to infinity.

4.1.3 Giving a better Asymptotic Development of $p(k)$

We can actually go further than Theorem 9 and give a more detailed description of the convergence of $p(k)$. As $p(k)$ is expected to converge to $\frac{\sqrt{2}}{2}$, consider the following change of variable, centring the variable in 0 and multiplying it by $\sqrt{n}$, by analogy to classical central limit theorems. Let

$$Y^{(n)}(k) = \sqrt{n}\left(p(k) - \frac{\sqrt{2}}{2}\right).$$

Note that none of these operations affect the status of $Y^{(n)}(k)$ as an homogenous Markov chain. We have

$$\mathbb{E}[Y^{(n)}(k+1) - Y^{(n)}(k)|Y^{(n)}(k)] = \sqrt{n}(\mathbb{E}[p(k+1) - p(k)|p(k)]).$$

Hence, from (4.4),

$$\mathbb{E}[Y^{(n)}(k+1) - Y^{(n)}(k)|Y^{(n)}(k)] = \frac{1}{\sqrt{n}}(1 - 2p(k)^2\frac{n}{n-1} + p(k)\frac{2}{n-1}).$$

Using $p(k) = \frac{\sqrt{2}}{2} + \frac{Y^{(n)}(k)}{\sqrt{n}}$, we get

$$\mathbb{E}[Y^{(n)}(k+1) - Y^{(n)}(k)|Y^{(n)}(k)] = \frac{\sqrt{2}-1}{\sqrt{n}(n-1)} + Y^{(n)}(k)(-\frac{2\sqrt{2}}{n-1} + \frac{2}{n(n-1)}) + Y^{(n)}(k)^2(-\frac{2}{\sqrt{n}(n-1)})$$

which yields the equivalent

$$n\mathbb{E}[Y^{(n)}(k+1) - Y^{(n)}(k)|Y^{(n)}(k)] \approx -2\sqrt{2}\ Y^{(n)}(k)$$

when n goes to infinity. We have

$$\mathbb{E}[(Y^{(n)}(k+1) - Y^{(n)}(k))^2|Y^{(n)}(k)] = n(\mathbb{E}[(p(k+1) - p(k))^2|p(k)]).$$

Hence, since $(p(k+1) - p(k))^2$ is always $\frac{1}{n^2}$, we get :

$$n\mathbb{E}[(Y^{(n)}(k+1) - Y^{(n)}(k))^2|Y^{(n)}(k)] = 1.$$

Set $a(x) = -2\sqrt{2}x$ and $b(x) = 1$. From the above calculations we have for all $R < +\infty$,

$$\lim_{n\to\infty} sup_{|x|\leq R}|a^{(n)}(x) - a(x)| = 0$$

$$\lim_{n\to\infty} sup_{|x|\leq R}|b^{(n)}(x) - b(x)| = 0.$$

Since the jumps of $Y^{(n)}$ are bounded in absolute value by $\frac{1}{\sqrt{n}}$, $\Delta_\epsilon^{(n)}$ is null, as soon as $\frac{1}{\sqrt{n}}$ is smaller than ϵ. So

$$\forall\epsilon > 0, \lim_{n\to\infty} sup_{|x|\leq R}\Delta_\epsilon^{(n)} = 0.$$

Moreover, we still have

$$\forall R < +\infty, \sup_{|x|\leq R} K^{(n)}(x) < \infty.$$

Now the stochastic differential equation

$$dX(t) = -2\sqrt{2}X(t)dt + dB(t) \tag{4.7}$$

is of a well-known type. Indeed, for $b > 0$ and $\sigma \neq 0$, stochastic differential equations of type

$$dX(t) = -bX(t)dt + \sigma dB(t)$$

are called *Langevin equations* [39]. Langevin equations are known to have a unique solution for all initial conditions $X(0) = x$. This solution is an *Orstein-Uhlenbeck* process, and is given by (see e.g., [22])

$$X(t) = e^{-bt}X(0) + \int_0^t e^{-b(t-s)}\sigma dB(s).$$

For all initial conditions $X(0)$, $X(t)$ is known to converge in law, when t goes to infinity, to the Gaussian distribution $\mathcal{N}(0, \frac{\sigma^2}{2b})$. This latter Gaussian distribution is an invariant distribution for the Orstein-Uhlenbeck process. See for example [22].

We now have all the ingredients to apply Theorem 8 yielding the following result:

Theorem 10 *For all t,*

$$p(\lfloor nt \rfloor) = \frac{\sqrt{2}}{2} + \frac{1}{\sqrt{n}}A_n(t),$$

where $(A_n(t))_{t\geq 0}$ converges in law to the unique solution of the stochastic differential equation (4.7)*.*

Since $(A_n(t))_{t\geq 0}$ converges in law to the unique solution $X(t)$ of the stochastic differential equation (4.7) and $X(t)$ converges in law to the Gaussian distribution $\mathcal{N}(0, \frac{\sigma^2}{2b})$, we will write

$$p(\lfloor nt \rfloor) \approx_{n,t\to\infty} \frac{\sqrt{2}}{2} + \frac{1}{\sqrt{n}}\mathcal{N}(0, \frac{\sigma^2}{2b}).$$

4.2 General Framework

Going back to the more general framework of a population of n agents taking states in any finite set Q, we will now show that a general result similar to Theorem 10 holds.

Transition rules of a population protocol are of the form:

$$q\ q' \to \delta_1(q, q')\ \delta_2(q, q')$$

for all $(q_1, q_2) \in Q^2$. Let us define the Markov chain $Y_i^{(n)}$ corresponding to the vector of $\mathbb{R}^Q$ whose components are the proportions of agents in the different states. Let

$$X^{(n)}(t) = Y^{(n)}_{\lfloor nt \rfloor} + (nt - \lfloor nt \rfloor)(Y^{(n)}_{\lfloor nt+1 \rfloor} - Y^{(n)}_{\lfloor nt \rfloor}).$$

Theorem 11 *Let b be the function defined by :*

$$b(x) = \sum_{(q,q')\in Q^2} x_q x_{q'}(-(e_q + e'_q) + e_{\delta_1(q,q')} + e_{\delta_2(q,q')})$$

where $(e_q)_{q\in Q}$ is the canonical base of $\mathbb{R}^Q$. Then for all sequences of initial conditions $Y_0^{(n)} \to x$, the sequence of random processes $X^{(n)}$ converges in law to the solution of the ordinary differential equation:

$$dX(t) = b(X(t))dt, \qquad X(0) = x. \tag{4.8}$$

Proof. Y_i^n is of the form required by Theorem 8, with $\pi^{(n)}(x,.)$ being the sum of $5^{|Q|}$ Dirac distributions : the variation of the proportion of agents in any given state belongs to $\{\frac{-2}{n}, \frac{-1}{n}, 0, \frac{1}{n}, \frac{2}{n},\}$ and the probabilities of any of these variations are clearly only dependent on the current state x.

Now let us define $a^{(n)}(x), b^{(n)}(x), K^{(n)}(x)$ and $\Delta_\epsilon^{(n)}$ as in Theorem 8. Let R be any finite non-negative real number. As in the example of Section 4.1, since, at any given time step, at most two out of n agents change state, $\Delta_\epsilon^n = 0$ if $\epsilon > \frac{4}{n}$ and thus

$$\lim_{n\to\infty} sup_{|x|\leq R}\Delta_\epsilon^{(n)} = 0, \forall\epsilon > 0.$$

Moreover,

$$\sup_{|x|\leq R} K^{(n)}(x) < \infty$$

still holds.

Similarly

$$\forall x \in \mathbb{R}^{|Q|}, |x| \leq R \Rightarrow |a^{(n)}(x)| \leq \frac{4|Q|}{n}.$$

So if we take $a(x) = 0$, we have

$$\lim_{n\to\infty} sup_{|x|\leq R}|a^{(n)}(x) - a(x)| = 0.$$

Let us write, for all $(q,q') \in Q^2, q \neq q'$,

$$\Pi_{q,q'}^{(n)}(x) = x_q x_{q'} \frac{n}{n-1}$$

and

$$\Pi_{q,q}^{(n)}(x) = x_q x_q \frac{n}{n-1} - \frac{x_q}{n-1}.$$

Then $\Pi_{q,q'}^{(n)}(x)$ is exactly the probability of an encounter between an agent in state q and an agent in state q' to happen when the population is in configuration x. We then have :

$$b^{(n)}(x) = \sum_{(q,q')\in Q^2} \Pi_{q,q'}^{(n)}(x)(-(e_q + e'_q) + e_{\delta_1(q,q')} + e_{\delta_2(q,q')}),$$

or

$$b^{(n)}(x) = \frac{n}{n-1}b(x) - \frac{1}{n-1}\sum_{q\in Q} x_q(-2e_q + e_{\delta_1(q,q)} + e_{\delta_2(q,q)}).$$

Thus, finally,

$$\lim_{n\to\infty} sup_{|x|\leq R}|b^{(n)}(x) - b(x)| = 0.$$

We can now conclude by Theorem 8. □

In view of Theorem 11, we get that the behavior of a protocol can be well approached by an ordinary differential equation, when the size of the population becomes large. In particular, if x^* is some stable equilibrium of the differential equation, then one expects $\overline{Y}^{(n)}(t)$ to converge to x^* whenever it starts close enough to x^*. Unfortunately, the notion of convergence involved in Theorem 11 (i.e., convergence in law) is too weak to derive this conclusion directly. And it is unclear that arguments similar to the asymptotic development of the toy example computing $\frac{\sqrt{2}}{2}$ can be generalized to any LPP.

However, we can generalize the theorem proven in [8], which allows us to show that if the ordinary differential equation in Theorem 11 has an exponentially stable equilibrium $b(x^*) = 0$, then, for every $\epsilon > 0$, and for every $0 < \mu < 1$, there is a neighbourhood U of x^* and some integers n and t, both polynomial in $1/\epsilon$, which guarantee that, with probability at least μ, we have $\|\overline{Y}^{(n)}(t) - x^*\| \leq \epsilon$ whenever the initial configuration belongs to U.

Let $F : K \subset \mathbb{R}^d \to \mathbb{R}^d$ be some $\mathcal{C}^1$ function over some compact K, and $x^* \in K$ such that :

- the ordinary differential equation (ODE)

$$\frac{dX}{dt} = F(X) \tag{4.9}$$

  over $K \subset \mathbb{R}^d$ is *locally convergent*: there is some neighbourhood U of x^*, such that for all ϵ, there is some $T(\epsilon)$ so that, whatever $X(0)$ is, any solution of the ODE is such that $||X(t) - x^*|| \leq \epsilon$ for $t \geq T(\epsilon)$.

- the ODE 4.9 is *efficiently locally convergent*, that is we have $T(\epsilon) \leq \mathcal{O}(\ln 1/\epsilon)$.

Let $(P_n(k))_k$ be a sequence of random variables taking values in compact K, and let c and d be two integers so that for all n and k, we have

$$P_n(k+1) - P_n(k) = \frac{1}{n}F(P_n(k)) + \frac{1}{n}\epsilon_n(k) + \frac{1}{n}\rho_n(k),$$

where $\epsilon_n(k)$ is a deterministic term taking value in $[-\frac{d}{n}, \frac{d}{n}]$, and $\rho_n(k)$ is random variable taking value in interval $[-c, c]$,

Theorem 12 (Generalization of Theorem 1 from [8]) *Let P_n be defined as above. Then for any $\epsilon > 0$ arbitrarily close to 0, and for probability μ arbitrary close to 1, there exist integers n and k that guarantee that whatever the initial condition $P_n(0) \in U$ is, we have with probability at least μ,*

$$||P_n(k) - x^*|| \leq \epsilon.$$

Moreover, whenever μ is fixed, $n = n(\epsilon)$ and $k = k(\epsilon)$ can be taken bounded by a polynomial in $1/\epsilon$.

The exact proof of Theorem 1 from [8] holds, provided we add the restriction that the system start in the attracting basin U of x^*, instead of accepting any initial configuration (as [8] considered the case of a single globally attractive equilibrium). We will however include the generalized proof for the sake of completeness.

Proof. Fix precision $\epsilon > 0$ and probability $0 < \mu < 1$.

Let X be a solution of ODE (4.9) with $X(0) \in U$. From Taylor-Lagrange on function X, we have for $T = \frac{k}{n}$,

$$X(T + \frac{1}{n}) - X(T) = \frac{1}{n}F(X(T)) + \frac{1}{2n^2}F'(\chi)F(\chi),$$

where $\chi \in [T, T + \frac{1}{n}]$. Let $\tilde{P}_n(\frac{k}{n}) = P_n(k)$, for all k, n. We can then write for $T = \frac{k}{n}$,

$$X(T + \frac{1}{n}) - \tilde{P}_n(T + \frac{1}{n}) = X(T) - \tilde{P}_n(T) + \frac{1}{n}\Big(F(X(T)) - F(\tilde{P}_n(T))\Big) - \frac{1}{n}\mu_n(T), \quad (4.10)$$

where $\mu_n(T) = \epsilon_n(k) + \rho_n(k) - \frac{1}{2n}F(\chi_n)F'(\chi_n)$. Summing (4.10) from 0 to k, yields

$$X(\frac{k+1}{n}) - \tilde{P}_n(\frac{k+1}{n}) = X(0) - \tilde{P}_n(0) + \sum_{i=0}^{k}\frac{1}{n}\left(F(X(\frac{i}{n})) - F(\tilde{P}_n(\frac{i}{n}))\right) - \sum_{i=0}^{k}\frac{1}{n}\mu_n(\frac{i}{n})$$

Since F is $\mathcal{C}^1$ over compact K, it is Λ-Lipschtiz for some $\Lambda > 0$.Using the fact that $X(0) - \tilde{P}_n(0) = 0$, and the fact that F is Λ-Lipschtiz, this yields

$$\left|X(\frac{k+1}{n}) - \tilde{P}_n(\frac{k+1}{n})\right| \leq \sum_{i=0}^{k}\frac{\Lambda}{n}\left|X(\frac{i}{n}) - \tilde{P}_n(\frac{i}{n})\right| + \left|\sum_{i=0}^{k}\frac{1}{n}\mu_n(\frac{i}{n})\right|$$

Denote $\theta_k = \sum_{i=0}^{k}\left|X(\frac{i}{n}) - \tilde{P}_n(\frac{i}{n})\right|$. This allows us to reformulate this as

$$\theta_{k+1} - \theta_k \leq \frac{\Lambda}{n}\theta_k + \left|\sum_{i=0}^{k}\frac{1}{n}\mu_n(\frac{i}{n})\right|$$

Recall Gronwall's Lemma :

Lemma 12 (Gronwall's Lemma: e.g. [24, page 213]) *Suppose that for some sequences $h_k, \theta_k \geq 0$ and $\epsilon_k \in \mathbb{R}$ we have $\theta_{k+1} \leq (1+\Lambda h_k)\theta_k + |\epsilon_k|$. Then*

$$\theta_k \leq e^{\Lambda(t_k - t_0)}\theta_0 + \sum_{0\leq i\leq k-1} e^{\Lambda(t_k - t_{i+1})}|\epsilon_i|,$$

where $t_{k+1} = t_k + h_k$, for all k.

We can apply Gronwall's Lemma, taking $h_k = \frac{1}{n}$, $\epsilon_k = \left|\sum_{i=0}^{k} \frac{1}{n}\mu_n(\frac{i}{n})\right|$. Yielding

$$\theta_k \leq \sum_{0\leq i\leq k-1} e^{\frac{\Lambda}{n}(k-i-1)} \left|\sum_{j=0}^{i} \frac{1}{n}\mu_n(\frac{j}{n})\right|,$$

and hence

$$\sup_{0\leq i\leq k}\left|X(\frac{i}{n}) - \tilde{P}_n(\frac{i}{n})\right| \leq \sum_{0\leq i\leq k-1} e^{\frac{\Lambda}{n}(k-i-1)} \left|\sum_{j=0}^{i} \frac{1}{n}\mu_n(\frac{j}{n})\right|. \tag{4.11}$$

This implies

$$\sup_{0\leq i\leq k}\left|X(\frac{i}{n}) - \tilde{P}_n(\frac{i}{n})\right| \leq \nu(k,n) \sup_{0\leq i\leq k} \left|\sum_{j=0}^{i} \frac{1}{n}\mu_n(\frac{j}{n})\right|, \tag{4.12}$$

where

$$\nu(k,n) = \sum_{0\leq i\leq k-1} e^{\frac{\Lambda i}{n}} = \frac{e^{\frac{\Lambda k}{n}} - 1}{e^{\frac{\Lambda}{n}} - 1} \leq e^{\frac{\Lambda k}{n}} \frac{1 - e^{-\frac{\Lambda k}{n}}}{1 - e^{-\frac{\Lambda}{n}}}$$

which is, doing an asymptotic development, in $\mathcal{O}(e^{\frac{\Lambda k}{n}})$, when n and $T = k/n$ are big enough, say when $n \geq n_1$ and $T \geq T_1$.

Decomposing $\mu_n(k/n) = \epsilon_n(k) + \rho_n(k) - \frac{1}{2n}F'(\chi_n)F(\chi_n)$, we obtain

$$\left|\sum_{j=0}^{i} \frac{1}{n}\mu_n(\frac{j}{n})\right| \leq \left|\sum_{j=0}^{i} \frac{1}{n}\epsilon_n(j)\right| + \left|\sum_{j=0}^{i} \frac{1}{n}\rho_n(j)\right| + \left|\sum_{j=0}^{i} \frac{1}{2n^2}F'(\chi_n)F(\chi_n)\right|$$

As $\epsilon_n(k)$ was assumed to take values in $[-\frac{d}{n}, \frac{d}{n}]$, the first term can be then bounded as follows

$$\left|\sum_{j=0}^{i} \frac{1}{n}\epsilon_n(j)\right| \leq \frac{d(i+1)}{n^2}.$$

The third term can be bounded as follows

$$\left|\sum_{j=0}^{i} \frac{1}{2n^2}F'(\chi_n)F(\chi_n)\right| \leq \frac{M_1 M_2(i+1)}{2n^2},$$

given that F is bounded on K, and that $F' = X$ is also bounded on K by respective constants M_1 and M_2.

Equation (4.12) then allows to write

$$\left| X(\frac{k}{n}) - \tilde{P}_n(\frac{k}{n}) \right| \leq \nu(k,n) \left(\frac{d(k+1)}{n^2} + \frac{M_1 M_2 (k+1)}{2n^2} \right) + \frac{\nu(k,n)}{n} \sup_{0 \leq i \leq k} \left| \sum_{j=0}^{i} \rho_n(j) \right|,$$

if one prefers

$$\left| X(\frac{k}{n}) - \tilde{P}_n(\frac{k}{n}) \right| \leq \mathcal{O}(e^{\Lambda T} T \frac{1}{n}) + \mathcal{O}(e^{\Lambda T} \frac{1}{n}) \sup_{0 \leq i \leq k} \left| \sum_{j=0}^{i} \rho_n(j) \right|$$

where $T = \frac{k}{n}$, when $n \geq n_1$ and $T \geq T_1$.

Recall now Azuma-Hoeffding's Inequality (see e.g. [44]): Let $Z_1, Z_2, \cdots, Z_n$ a martingale such that

$$|Z_k - Z_{k-1}| \leq c_k.$$

Then for all $t \geq 0$ and all $\lambda > 0$,

$$Pr(|Z_t - Z_0| \geq \lambda) \leq 2e^{-\lambda^2/(2\sum_{k=1}^{t} c_k^2)}.$$

Now consider $Z_0 = 0$, and $Z_k = \sum_{j=0}^{k-1} \rho_n(j)$ for $k > 0$. By hypothesis $|Z_k - Z_{k-1}| \leq c$. So, for all $\lambda > 0$,

$$Pr(|Z_k| \geq \lambda) \leq 2e^{-\lambda^2/(2kc^2)}.$$

Using some union bounds,

$$Pr(\sup_{0 \leq i \leq k} |\sum_{j=0}^{i} \rho_n(j)| > \lambda) \leq P(\bigcup_{0 \leq i \leq k} |\sum_{j=0}^{i} \rho_n(j)| > \lambda) \leq \sum_{i=0}^{k} P(|\sum_{j=0}^{i} \rho_n(j)| > \lambda),$$

which is less than

$$\sum_{i=0}^{k} 2e^{-\lambda^2/(2(i+1)c^2)} \leq 2(k+1)e^{-\lambda^2/(2c^2)}.$$

Fix κ so that $1 - 2(k+1)e^{-\kappa^2 T^2} \geq \mu$ whenever $n \geq n_1$ and $k \geq k_1 = T_1 n_1$. Take then $\lambda = \kappa T c \sqrt{2}$. With probability more than μ

$$\sup_{0 \leq i \leq k} |\sum_{j=0}^{i} \rho_n(j)| \leq \kappa c \sqrt{2} T,$$

hence

$$\left| X(\frac{k}{n}) - \tilde{P}_n(\frac{k}{n}) \right| \leq \mathcal{O}(e^{\Lambda T} T \frac{1}{n}) + \mathcal{O}(e^{\Lambda T} T \frac{1}{n}) = \mathcal{O}(e^{\Lambda T} T \frac{1}{n})$$

Take any $T \geq \max(T(\frac{\epsilon}{2}), T_1)$ so that

$$||X(T) - x^*|| \leq \frac{\epsilon}{2}.$$

Then take any $n \geq n_1$ where n is big enough so that $\mathcal{O}(e^{\Lambda T} T \frac{1}{n}) \leq \frac{\epsilon}{2}$: as n_1 is some constant (not depending on ϵ) n can be taken in $\mathcal{O}(\frac{1}{\epsilon} e^{\Lambda T} T) = \mathcal{O}(\frac{1}{\epsilon}(\frac{1}{\epsilon})^{\mathcal{O}(1)} \ln \frac{1}{\epsilon})$, that is to say, polynomial in $\frac{1}{\epsilon}$. Consider then $k = \max(k_1, Tn)$. We have

$$||P_n(k) - x^*|| = ||\tilde{P}_n(\frac{k}{n}) - x^*|| \leq ||\tilde{P}_n(\frac{k}{n}) - X(\frac{k}{n})|| + ||X(\frac{k}{n}) - x^*|| \leq \epsilon :$$

as k_1 is some constant (not depending on ϵ), and as n is polynomial in $\frac{1}{\epsilon}$, k can also be taken as polynomial in $\frac{1}{\epsilon}$. □

Corollary 2 *If x^* is an exponentially stable equilibrium of the ordinary differential equation from Theorem 11, there exists a neighbourhood U of x^* such that, for any given probability $\mu, 0 < \mu < 1$, any margin $\epsilon > 0$, there exist $n(\mu, \epsilon)$, and $t(\mu, \epsilon$ such that, a population protocol executing on a population of size at least $n(\mu, \epsilon)$, then after time $t(\mu, \epsilon)$, the population is at distance at most ϵ of x^* with probability at least μ.*

Proof. Simply apply Theorem 12 with $P_n(k) = p_n(k)$, the proportion vector, $F(x) = b(x)$. Then $\epsilon_n(k) = b_n(p_n(k)) - b(p_n(k))$, is the deterministic perturbation and $\rho_n(k)$ the randomized perturbation is defined by $\rho_n(k) = n\,(p_n(k+1) - p_n(k)) - b_n(p_n(k)$. □

4.3 Computing with LPPs.

We have now all ingredients sufficient to formally define computing with LPPs.

Let $\mathcal{P} = (Q, \Delta)$ be a LPP. A vector of real numbers $x^* = (x_1, \ldots, x_{|Q|}) \in [0,1]^{|Q|}$ is said to be an *equilibrium* of a $\mathcal{P}$ if and only if $b(x^*) = 0$. That is to say, the constant solution $f(t) = (x_1, \ldots, x_{|Q|})$ is a fix-point solution of the differential equation in Lemma 11. An equilibrium x^* of $\mathcal{P}$ is said to be *stable* if it is the (exponentially) stable equilibrium of the associated ordinary differential equation. In other words, there is a neighbourhood U of the equilibrium x^* such that any trajectory starting from U converges exponentially fast to the equilibrium. This is equivalent to saying that the eigenvalues of the Jacobian matrix of b in x^* has negative real parts [32].

Definition 9 *A real number ν is said to be computable by LPP if there exists a vector $x^* = (x_1, x_2, ..., x_k) \in [0,1]^k$ such that $\sum_{i=1}^{k} x_i = 1$, and a LPP $\mathcal{P}$, admitting finitely many equilibria, such that $(x_1, x_2, ..., x_k)$ is a stable equilibrium of $\mathcal{P}$ and $\sum_{q_i \in Q^+} x_i = \nu$ where Q^+ is the set of marked states for $\mathcal{P}$.*

Notice that the Definition 9 requires the system to have finitely many equilibria. This assumption is mainly to avoid pathological cases, in particular the case of idle systems $q\ q' \to q\ q'$ for all q, q'. Indeed, in idle systems, all initial states are equilibria, and such a system could compute any real of $[0,1]$, depending on the initial configuration.

4.4 Conclusion

In this chapter, we constructed a natural extension of population protocols aiming at capturing the specifics of very large populations called Large-Population Protocols. Illustrating how such LPPs behave on a toy example, we used this analysis to define a computation a number ν by a LPP $\mathcal{P}$ to be the convergence to ν of the fraction of agents in a subset of marked states, when both the size of the population and time go to infinity. We also showed that if a protocol $\mathcal{P}$ computes a real number ν, then ν can be approximated within a margin of $\epsilon > 0$ with high probability with a population of size n and after time t where n and t are both polynomial in $\frac{1}{\epsilon}$, starting from an initial configuration $X(0)$ within a neighbourhood U of the equilibrium x^* associated with ν.

The next step is, naturally, to determine what numbers can be computed by LPPs. This is the object of chapter 5.

Chapter 5

The computational power of LPPs

In this chapter, we will establish our main result on Large Population Protocols:

Theorem 13 *A real number $\nu \in [0,1]$ is computable by an LPP if and only if it is algebraic.*

That a computable number ν must be algebraic is a consequence of ν being an equilibrium of the ODE approximating the protocol and of the polynomial structure of such an ODE. Reciprocally we show that all algebraic numbers are indeed computable by showing how to construct a protocol to compute any given algebraic number ν. To simplify this construction, we first prove that one can consider the transition rules of LPPs to be probabilistic without changing the computational power as long as the probabilities involved are rational fractions. This result is achieved by using the inherent randomness of the system to emulate a probabilistic protocol with a deterministic one. Then we construct a probabilistic protocol computing a given algebraic number ν. These results are to appear in [16].

5.1 Any computable number is algebraic.

We first prove that there is an intrinsic limitation to the power of LPPs, namely not a single transcendental number can be computed by LPPs. This limitation is essentially a direct consequence of arguments from model theory (mainly Tarski's effective procedure for quantifier elimination over real closed fields).

Lemma 13 *For every $\nu \in [0,1]$, if ν is computable by a LPP then ν is algebraic.*

Proof. The impossibility of computing transcendent numbers follows from the fact that, by definition, a computable real number ν must correspond, in the case where only one state is marked, to an equilibrium $x^* = (\nu, x_2, ..., x_k) \in [0,1]^k$ of an ordinary differential equation of the form $dX(t) = b(X(t))dt$, where b is a component-wise polynomial function. Therefore $b(x^*) = 0$. (The case where several states are marked can be treated similarly). Since function b is, component-wise, polynomial, we get that x^* is solution of a system of polynomial equations.

The lemma then follows from the following claim: any solution of a system of polynomial equations which has finitely many solutions is, component-wise, algebraic.

This claim can be established using arguments from Model theory. It essentially follows from Tarski's effective procedure for quantifier elimination over real closed fields (see e.g. [19]). More specifically, the field $\mathbb{R}$ of real numbers, as well as the field $\mathbb{Q}_{alg}$ of algebraic numbers over $\mathbb{Q}$, are known to be real closed fields. Now, real closed fields are known to be elementary equivalent, that is they satisfy the same first-order properties. In other words, any first order predicate (in the first-order language of fields) is true in one real closed field if and only if it is true in another. In particular, concerning $\mathbb{R}$ and $\mathbb{Q}_{alg}$, since "having a fixed finite number of solutions" is expressible in first order logic, if a system of polynomial equations has a fixed finite number of solutions over $\mathbb{R}$, then it must also have a fixed finite number of solutions over $\mathbb{Q}_{alg}$. Since $\mathbb{Q}_{alg}$ contains only algebraic numbers, each of these finitely many solutions is algebraic. Therefore, each component of each solution is the root of some minimal polynomial with rational coefficients. Now, "S is a solution of a system of polynomial equations implies that each component of S is a root of the corresponding polynomials" is a first order formula that is true over $\mathbb{Q}_{alg}$. Therefore it holds also over $\mathbb{R}$. In other words, all components of the solutions of the system of polynomial equations are algebraic. $\square$

5.2 Computing Algebraic Numbers

The remaining part of the chapter is entirely dedicated to proving that every algebraic number is indeed computable by an LPP. The proof is constructive, meaning that we describe how to construct a LPP computing ν, for any given algebraic number $\nu \in [0,1]$. The construction of the protocol is made in four stages, corresponding to the following four sections. The first stage consists in the design of LPPs computing rational numbers. The second stage consists in using the computation of rational numbers as a subroutine for the emulation of probabilistic transition rules. This stage will allow us to consider LPPs with transition rules of the form

$$q_i \; q_j \to \alpha_{i,j,k,l} \; q_k \; q_l$$

to be understood as: the interaction between two agents in respective states q_i and q_j results in the two agents moving to respective states q_k and q_l with probability $\alpha_{i,j,k,l}$. Then, the third stage of our proof is the construction of a (probabilistic) protocol $\mathcal{P}$ admitting ν as an equilibrium. We assume given a degree-δ polynomial $P \in \mathbb{Q}[X]$ with root ν. The protocol $\mathcal{P}$ is based on one specific choice for another degree-δ polynomial $P' \in \mathbb{Q}[X]$, and, essentially, satisfies that $(x_1, \ldots, x_\delta) \in [0,1]^\delta$ is an equilibrium of $\mathcal{P}$ if and only if (1) $P'(x_1) = 0$, (2) $x_i = x_1^i$ for every $1 \leq i < \delta$, and (3) $x_\delta = 1 - \sum_{i=1}^{\delta} x_i$. Finally, the fourth stage of the construction consists in proving that we can actually enforce this protocol $\mathcal{P}$ to be stable near the equilibrium.

5.2.1 Computing Rationals

Lemma 14 *Let $\nu \in [0,1]$ be a rational number. There exists a LPP computing ν.*

Proof. We first show that, for every integer $k \in \mathbb{N}$, there exists a protocol that, given any initial configuration, converges to the unique equilibrium $(\frac{1}{k}, \ldots, \frac{1}{k})$. For this purpose, consider the protocol $\mathcal{M}$ over states $Q_k = \{1, \ldots, k\}$ given by the following transition rules: $i\ j \to (i+1)\ (j+1)$ where, for $q \in Q_k$, $q+1$ stands for $(q \bmod k)+1$. The dynamic system describing this protocol is

$$\frac{dx_i}{dt} = 2(x_{i-1} - x_i).$$

If $f : \mathbb{R} \to [0,1]^k$ is a solution of this differential system, then, considering $g(t) = \|f(t) - (\frac{1}{k}, \ldots, \frac{1}{k})\|^2$, where $\|x\|$ is the Euclidean norm of vector x, we get

$$\frac{dg(t)}{dt} = 4 \sum_{i=1}^{k} x_i(x_{i-1} - x_i).$$

A simple induction on $k \in \mathbb{N}$ enables to show that $\frac{dg(t)}{dt} \leq 0$ for every vector $x \in [0,1]^k$, and $\frac{dg(t)}{dt} = 0$ if and only if $x_1 = x_2 = \ldots = x_k$, thereby proving that f converges to $(\frac{1}{k}, \cdots, \frac{1}{k})$ when $t \to \infty$. This, in particular, guarantees that $(\frac{1}{k}, \ldots, \frac{1}{k})$ is the only stable equilibrium of $\mathcal{M}$.

Now, let $\nu = p/q \in \mathbb{Q}$. Computing ν is achieved by using $\mathcal{M}$ as above, with $k = q$, and setting marked states as the first p states of Q. $\square$

5.2.2 Derandomization

We now prove that considering LPPs with probabilistic transition rules where the probabilities are rational does not change the computing power of LPPs. Note that this result has its own independent interest. It essentially says that the random choice of the agents involved in the transition provides enough randomness, and that there are no further benefits from using randomization in the transition rules. Nevertheless, using probabilistic LPPs, or PLPPs for short, considerably simplifies the construction of LPPs computing algebraic numbers.

We focus on PLPPs, where the transition rules are defined by

$$q_i\ q_j \to \alpha_{i,j,k,l}\ q_k\ q_l$$

where all $\alpha_{i,j,k,l}$ are rational numbers. Recall that such probabilistic transition rules mean that an interacting pair of agents in respective states q_i and q_j will move to respective states q_k and q_l, with probability $\alpha_{i,j,k,l}$. Of course, for such rules to be well-defined, we assume that for every pair $(q_i, q_j) \in Q^2$, we have

- for every $(q_k, q_l) \in Q^2$, $\alpha_{i,j,k,l} \geq 0$, and
- $\sum_{(q_k,q_l)\in Q^2} \alpha_{i,j,k,l} = 1$.

Notice that all previous definitions and statements can be easily extended to PLPPs. In particular, Theorem 11 and Theorem 12 still hold, by replacing function b in Lemma 11 by

$$b(x) = \sum_{(q_1,q_2)\in Q^2} x_{q_1}x_{q_2}\Big(- e_{q_1} - e_{q_2} + \sum_{(q_3,q_4)\in Q^2} \alpha_{q_1,q_2,q_3,q_4}(e_{q_3} + e_{q_4})\Big)$$

Now we prove that probabilistic LPPs (with rational probabilities) are actually not more powerful than classical LPPs

Lemma 15 *Let $\nu \in [0,1]$, and assume that there exists a probabilistic LPP computing ν, with rational probabilities. Then there exists a (deterministic) LPP computing ν.*

Proof. Let $\mathcal{P}$ be a probabilistic LPP computing ν, with rational probabilities. Let us denote $Q = \{q_1, \dots, q_{|Q|}\}$ the set of states for protocol $\mathcal{P}$, and let $\tilde{\nu} = (\nu_1, \cdots, \nu_d)$ be some associated stable equilibrium. Let $\{\alpha_{i,j,k,l}, 1 \le i,j,k,l \le |Q|\}$ be the family of rational coefficients of its probabilistic transition rules

$$q_i\ q_j \to \alpha_{i,j,k,l}\ q_k\ q_l.$$

Let $m \in \mathbb{N}$ be a common denominator of all $\alpha_{i,j,k,l}$. A natural way to implement such a probabilistic random transition between two agents in states $q_i\ q_j$ would be to generate a random number $r \in \{1, \dots, m\}$. Then, using the $\alpha_{i,j,k,l}$ to partition $[0,1]$, one could decide which transition to apply according to where $\frac{r}{m}$ falls in this partition. If r is generated uniformly at random in $\{1, \dots, m\}$, this approach would yield the desired result. The idea of our proof is the same except that we use the protocol $\mathcal{P}_m$ described in Lemma 14 to generate r. Using the same notations as the ones introduced in Lemma 14, we construct a new protocol over states $Q \times Q_m$. For every pair $(q_i, q_j) \in Q^2$, the family $\{\alpha_{i,j,k,l},\ (q_k, q_l) \in Q^2\}$ defines a partition of $[0,1]$. For any integer $r \in \{1, \dots, m\}$, if (q_k, q_l) is the pair of Q^2 such that $\frac{r}{m}$ is in the part of $[0,1]$ associated to $\alpha_{i,j,k,l}$ in the aforementioned partition, we define the transition rule interaction between agents in states (q_i, r) and (q_j, r') (where r' is any integer in Q_m) to be:

$$(q_i, r)(q_j, r') \to (q_k, r+1)(q_l, r'+1).$$

Intuitively, by considering solely the second element of every pair, the protocol executes $\mathcal{P}_m$ independently from what happens to the first element. Thus any equilibrium of $\tilde{\mathcal{P}}$ will, as previously, have agents equidistributed in Q_m, as far as the second part of their state is concerned. More precisely, the dynamics of the differential system corresponding to the obtained system over $Q \times Q_m$ are given by

$$\frac{dx(t)}{dt} = B(x) = (b_{1,1}(x), \cdots, b_{|Q|,1}(x), \cdots, b_{1,m}(x), \cdots, b_{|Q|,m}(x)),$$

where

$$b_{I,R}(x) = -2x_{I,R} + \sum_{i,j} x_{i,R-1}\tilde{x_j}\delta^{1,R}_{i,j,I} + \sum_{i,j} x_{j,R-1}(\sum_r x_{i,r}\delta^{2,r}_{i,j,I}) \tag{5.1}$$

and

- $\tilde{x}_i = \sum_{r=1}^m x_{i,r}$,
- $\delta^{1,R}_{q,q',I} = 1$ iff $\frac{R-1}{m}$ falls in the part of $[0,1]$ associated with $\alpha_{q,q,'I,q''}$ for some q'', and
- $\delta^{2,r}_{q,q',I} = 1$ iff $\frac{r}{m}$ falls in the part of $[0,1]$ associated with $\alpha_{q,q',q'',I}$ for some q''.

By construction, for any q, q', R, there is exactly one I such that $\delta^{1,R}_{q,q',I} = 1$. Similarly, for any q, q', r, there is exactly one I with $\delta^{2,r}_{q,q',I} = 1$. Since ν must satisfy $b_I(\nu) = 0$ where

$$b_I(y) = -2y_I + \sum_{i,j,l} y_i y_j \alpha_{i,j,I,l} + \sum_{i,j,k} y_i y_j \alpha_{i,j,k,I}$$

we get that

$$\nu^* = (\frac{\nu_1}{m}, \cdots, \frac{\nu_1}{m}, \cdots, \frac{\nu_{|Q|}}{m}, \cdots, \frac{\nu_{|Q|}}{m})$$

satisfying $b(\nu^*) = 0$. Thus, it is an equilibrium of the considered system over $Q \times Q_m$.

Concerning stability, let us denote $\overline{x}_R = \sum_i x_{i,R}$. Then

$$g(t) = \sum_R (\overline{x}_R(x(t)) - \frac{1}{m})^2$$

is always decreasing along the trajectory since

$$\frac{d}{dt} g(t) = \sum_R \left(\sum_I b_{I,R}(x(t)) \right) \left(\overline{x}_R(t) - \frac{1}{m} \right)$$

and since it can be checked that

$$\forall R, \sum_I b_{I,R}(x(t)) = 2(\overline{x}_{R-1} - \overline{x}_R).$$

As $g(t)$ is null if and only if $\overline{x} = (\frac{1}{m}, \cdots \frac{1}{m})$, it acts as a Liapunov function [32] for the dynamic. We get that $\overline{x}(t)$ must go to $(\frac{1}{m}, \cdots, \frac{1}{m})$ exponentially fast. Since $\tilde{\nu}$ is an exponentially stable equilibrium, the system $y' = b(y)$ must locally admit some Liapunov function $\tilde{L}(y(t))$, i.e., a function such that

$$\frac{d}{dt} \tilde{L}(y(t)) = \text{grad } \tilde{L}(y) \cdot b(y) \leq 0$$

with equality only in $y = \tilde{\nu}$. We can furthermore assume

$$\frac{d}{dt} \tilde{L}(y(t)) \leq -\alpha \|y(t) - \tilde{\nu}\|^2$$

for some positive α (see, e.g., [32]).

Some basic computations show that $|\sum_R b_{I,R} - b_I|$ can locally be bounded by $\beta \|\overline{x} - (\frac{1}{m}, \cdots, \frac{1}{m})\|^2$ for some constant β, for all I.

Then $L(x) = \tilde{L}(\tilde{x}_1, \cdots, \tilde{x}_{|Q|}) + K\|\overline{x} - (\frac{1}{m}, \cdots, \frac{1}{m})\|^2$ must locally be a Liapunov function of the whole system over $Q \times Q_m$, if K is chosen large enough. Indeed, we can always guarantee

$$\text{grad } L(x).B(x) \leq \text{grad } \tilde{L}(\tilde{x}).b(\tilde{x}) + \beta'\|\overline{x} - (\frac{1}{m}, \cdots, \frac{1}{m})\|^2$$

on a suitable neighborhood of ν^*, for some constant β'. It follows that

$$\frac{d}{dt}L(x(t)) \leq -\alpha\|\tilde{x}(t) - \tilde{\nu}\|^2 + (\beta' - K\gamma)\|\overline{x} - (\frac{1}{m}, \cdots, \frac{1}{m})\|^2$$

observing that $\frac{d}{dt}g(t)$ is less than $-\gamma\|\overline{x} - (\frac{1}{m}, \cdots, \frac{1}{m})\|^2$ for some positive constant β using eq. 5.4 above. This guarantees that ν^* is locally exponentially stable equilibrium, if K is taken large enough. □

5.2.3 Constructing Equilibria

In view of the previous two subsections, one can freely use probabilistic LPPs, whenever the probabilities are rational, in order to compute any algebraic number $\nu \in [0, 1]$. In this subsection, we will not yet produce a probabilistic LPP computing an algebraic number ν, as we will ignore stability which is only discussed in the next subsection, and solely focus on constructing a protocol with ν as an equilibrium. The aim of this subsection is actually to prove the following result:

Lemma 16 *For every algebraic number $\nu \in [0, 1]$, there exist $\delta \in \mathbb{N}$, $\lambda \in \mathbb{Q}$, and a protocol $\mathcal{P}$ such that $(\nu, \lambda\nu^2, \lambda^2\nu^3, \ldots, \lambda^{\delta-2}\nu^{\delta-1}, 1 - \sum_{i=1}^{\delta-1} \lambda^{i-1}\nu^i)$ is an equilibrium of $\mathcal{P}$.*

Proof. The lemma is in fact a consequence of the following result. Let $\nu \in (0, 1]$ be an algebraic number, and let $P(X) = \sum_{i=0}^{\delta} a_i X^i, P \in \mathbb{Q}[X]$, be a polynomial such that $P(\nu) = 0$, and $P(0) > 0$. We claim that there exist a rational number $\epsilon \neq 0$, and a protocol $\mathcal{P}_\epsilon$ with equilibrium $(\nu, \lambda\nu^2, \lambda^2\nu^3, \ldots, \lambda^{\delta-2}\nu^{\delta-1}, 1 - \sum_{i=1}^{\delta-1} \lambda^{i-1}\nu^i)$ described by the following differential equations:

$$\begin{cases} dx_1 &= \epsilon(a_0 + a_1x_1 + \sum_{i=2}^{\delta-1} \frac{a_{i+1}}{\lambda^{i-1}} x_{i-1}x_1) \\ dx_i &= \lambda x_1 x_{i-1} - x_i \text{ for every } i = 2, \ldots, \delta - 1 \\ dx_\delta &= -\sum_{i=1}^{\delta-1} dx_i. \end{cases}$$

where λ is a rational number such that $\lambda > 0$, and $\sum_{i=1}^{\delta-1} \lambda^{i-1}\nu^i \leq 1$. To establish that claim, we explicitly construct a protocol $\mathcal{P}_\epsilon$ over set of states $Q = \{1, \ldots, \delta\}$ with 1 serving as our marked state. Fix $\lambda \in \mathbb{Q}$, $\lambda > 0$ small enough, so that $\sum_{i=1}^{\delta-1} \lambda^{i-1}\nu^i \leq 1$. Then let

$$M = \max\Big(\{|\frac{a_{i+1}}{\lambda^{i-1}} + 2a_0 + a_1|, i \in \{2, ..., \delta - 1\}\} \cup \{|a_2 + a_0 + a_1|, |2a_0 + a_1|, a_0\}\Big)$$

and fix $\epsilon \in \mathbb{Q}$, $0 < \epsilon < \frac{1}{M}(1 - \frac{\lambda}{2})$. We define the family $(\alpha_{i,j,k,l})_{1 \leq i,j,k,l \leq \delta}$, that yields the transition rules for the protocol $\mathcal{P}_\epsilon$ as follows:

$$\begin{cases} i=1, j=1 & \Longrightarrow & \alpha_{1,1,1,1} = \epsilon\, \frac{a_2+a_1+a_0}{2} + \frac{1}{2} \text{ and } \alpha_{1,1,2,2} = \frac{\lambda}{2} \\ i=1, 1<j<\delta-1 & \Longrightarrow & \alpha_{1,j,1,1} = \epsilon\, \frac{\frac{a_{j+1}}{\lambda^{j-1}}+2a_0+a_1}{4} + \frac{1}{2} \text{ and } \alpha_{1,j,j+1,j+1} = \frac{\lambda}{4} \\ i=1, j=\delta-1 & \Longrightarrow & \alpha_{1,j,1,1} = \epsilon\, \frac{\frac{2a_\delta}{k^{\delta-2}}+2a_0+a_1}{4} + \frac{1}{2} \\ i=1, j=\delta & \Longrightarrow & \alpha_{1,j,1,1} = \epsilon\, \frac{2a_0+a_1}{4} + \frac{1}{2} \\ 1<i<\delta-1, j=1 & \Longrightarrow & \alpha_{i,1,1,1} = \epsilon\, \frac{\frac{a_{i+1}}{\lambda^{i-1}}+2a_0+a_1}{4} + \frac{1}{2} \text{ and } \alpha_{i,1,i+1,i+1} = \frac{\lambda}{4} \\ i=\delta-1, j=1 & \Longrightarrow & \alpha_{i,1,1,1} = \epsilon\, \frac{\frac{2a_\delta}{k^{\delta-2}}+2a_0+a_1}{4} + \frac{1}{2} \\ i=\delta, j=1 & \Longrightarrow & \alpha_{i,1,1,1} = \epsilon\, \frac{2a_0+a_1}{4} + \frac{1}{2} \\ i>1, j>1 & \Longrightarrow & \alpha_{i,j,1,1} = \epsilon\, \frac{a_0}{2} \end{cases}$$

And, for all $(k,l) \neq (\delta,\delta)$ not explicited above, we set $\alpha_{i,j,k,l} = 0$. Finally, if $(k,l) = (\delta,\delta)$, then $\alpha_{i,j,\delta,\delta} = 1 - \sum_{(k,l)\neq(\delta,\delta)} \alpha_{i,j,k,l}$.

By definition of M and ϵ, it follows that, for any pair (i,j), if $(k,l) \neq (\delta,\delta)$, then $0 \leq \alpha_{i,j,k,l} \leq 1$. Moreover, we have $0 \leq \sum_{(k,l)\neq(\delta,\delta)} \alpha_{i,j,k,l} \leq 1$. Thus, for every (i,j), $0 \leq \alpha_{i,j,\delta,\delta} \leq 1$. Therefore, the family $(\alpha_{i,j,k,l})$ properly defines a protocol $\mathcal{P}_\epsilon$. We now show that this protocol satisfies our needs. By construction, the dynamic of $\mathcal{P}_\epsilon$ is captured by the following system :

$$\forall k \in \{1,\ldots,\delta\},\ dx_k = \sum_{l=1}^{\delta} \sum_{i,j} (\alpha_{i,j,k,l} + \alpha_{i,j,l,k}) x_i x_j - x_k.$$

Now, if $k \neq l$ then $\alpha_{i,j,k,l} = 0$. Thus $dx_k = \sum_{i,j} 2\alpha_{i,j,k,k}\ x_i x_j - x_k$. In particular, for $1 < k < \delta$, we have

$$dx_k = \lambda x_1 x_{k-1} - x_k. \tag{5.2}$$

For the last component, we have $dx_\delta = -\sum_{i=1}^{\delta-1} dx_i$. Then computing dx_1 yields:

$$\begin{aligned} dx_1 &= \epsilon(a_2+a_0+a_1)x_1^2 + \sum_{i=2}^{\delta-1} \left(\epsilon\left(\frac{a_{i+1}}{\lambda^{i-1}} + 2a_0 + a_1\right) + 1\right) x_1 x_i + \sum_{i=2}^{\delta}\sum_{j=2}^{\delta} \epsilon a_0 x_i x_j + (2\epsilon a_0 + \epsilon a_1 + 1) x_1 x_\delta \\ &= \epsilon\Big(a_0 + a_1 x_1 + \sum_{i=2}^{\delta-1} \frac{a_{i+1}}{\lambda^{i-1}} x_{i-1} x_1\Big) + \Big(\sum_{i=1}^{\delta} x_i - 1\Big)\Big(\epsilon a_0 + \epsilon a_0 \sum_{i=1}^{\delta} x_i + (\epsilon a_1 + 1) x_1\Big) \end{aligned}$$

Since $\sum x_i = 1$, the last term of that equation is zero. Thus, for $\epsilon \leq \frac{1}{M}$, we get

$$dx_1 = \epsilon\ \Big(a_0 + a_1 x_1 + \sum_{i=2}^{\delta-1} \frac{a_{i+1}}{\lambda^{i-1}} x_{i-1} x_1\Big). \tag{5.3}$$

Equations 5.2 and 5.3 prove the claim, which completes the proof of Lemma 16. □

5.2.4 Enforcing Stability

Perhaps surprisingly, stability does not come for free, and the construction of the previous subsection is not sufficient to conclude the computation. One needs to enforce stability. For that purpose, the protocol of the previous subsection is modified in order to satisfy the stability criteria from the theory of dynamic systems. We prove the following result, which completes the proof of Theorem 13.

Lemma 17 *For every algebraic number $\nu \in [0,1]$, there exist $\delta \in \mathbb{N}$, $\lambda \in \mathbb{Q}$, and a protocol $\mathcal{P}$ such that $(\nu, \lambda\nu^2, \lambda^2\nu^3, \ldots, \lambda^{\delta-2}\nu^{\delta-1}, 1 - \sum_{i=1}^{\delta-1} \lambda^{i-1}\nu^i)$ is a* stable *equilibrium of $\mathcal{P}$.*

Proof. Let $\nu \in [0,1]$ be an algebraic number. Let $P = \sum_{i=0}^{\delta} a_i X^i$ be a polynomial in $\mathbb{Q}[X]$ such that $P(\nu) = 0$, and $P(0) = a_0 > 0$. Let P' be the derivative of P. If $P'(\nu) > 0$, then we consider the polynomial $Q(X) = -(X - \alpha)P(X)$ where $\alpha \in \mathbb{Q}$ satisfies $0 < \alpha < \nu$. Indeed, $Q'(\nu) = -(\nu - \alpha)P'(\nu) < 0$, and $Q(0) = \alpha P(0) > 0$. So, we can assume without loss of generality that $P'(\nu) < 0$. Let $\epsilon > 0$ be a rational number, and let us consider the protocol $\mathcal{P}_\epsilon$ constructed in the proof of Lemma 16, using polynomial ϵP. Recall that the behavior of this protocol is determined by the following differential equations:

$$dx_1 = \epsilon(a_0 + a_1 x_1 + \sum_{i=2}^{\delta-1} \frac{a_{i+1}}{\lambda^{i-1}} x_{i-1} x_1) \tag{5.4}$$

and, for $1 < k < \delta$, $dx_k = \lambda x_1 x_{k-1} - x_k$. By construction,

$$(\nu, \lambda\nu^2, \lambda^2\nu^3, \ldots, \lambda^{\delta-2}\nu^{\delta-1}, 1 - \sum_{i=1}^{\delta-1} \lambda^{i-1}\nu^i)$$

is an equilibrium of $\mathcal{P}_\epsilon$. We establish the stability of this equilibrium using standard techniques of Stability theory when studying non-linear autonomous systems. Specifically, we focus on the sign of the real-part of the eigenvalues of the Jacobian matrix, using the Routh-Hurwitz criterion.

From now on, we will only express the behavior of the protocol according to the $\delta - 1$ first variables, ignoring x_δ. Indeed, since $\sum_{i=1}^{\delta} x_i = 1$, we can restrict ourselves to the study of the system

$$\{(x_1, \ldots, x_{\delta-1}) \in [0,1]^{\delta-1}, \sum_{i=1}^{\delta-1} x_i \leq 1\}$$

after elimination of the last variable. Indeed, Equation 5.4 is independent from x_δ, and thus this substitution does not alter the equations of the system. A sufficient condition for such an equilibrium to be stable is that all the eigenvalues of J_ϵ have negative real part, where

$$J_\epsilon = J_\epsilon(\nu, \lambda\nu^2, \ldots, \lambda^{\delta-2}\nu^{\delta-1})$$

is the Jacobian matrix of the dynamic in this point. Specifically:

$$J_\epsilon = \begin{pmatrix} \epsilon(a_1 + 2a_2\nu + \sum_{i=3}^{\delta} a_i\nu^{i+1}) & \epsilon\frac{a_3\nu}{\lambda} & & \dots & \epsilon\frac{a_\delta\nu}{\lambda^{\delta-2}} \\ 2\lambda\nu & -1 & & 0 & \\ \lambda^2\nu^2 & \lambda\nu & -1 & & 0 \\ \vdots & 0 & \ddots & \ddots & 0 \\ \lambda^{\delta-2}\nu^{\delta-2} & 0 & & \lambda\nu & -1 \end{pmatrix}$$

We prove that there exists ϵ small enough such all the eigenvalues of J_ϵ have negative real part. By definition, the eigenvalues of J_ϵ are the roots of χ_ϵ, where χ_ϵ is the characteristic polynomial of J_ϵ defined by the following determinant:

$$\chi_\epsilon = |XI_{\delta-1} - J_\epsilon| = \begin{vmatrix} X - \epsilon(a_1 + 2a_2 r + \sum_{i=3}^{\delta} a_i\nu^{i+1}) & -\epsilon\frac{a_3\nu}{\lambda} & & \dots & -\epsilon\frac{a_\delta\nu}{\lambda^{\delta-2}} \\ -2\lambda\nu & X+1 & & 0 & \\ -\lambda^2\nu^2 & -\lambda\nu & X+1 & & 0 \\ \vdots & 0 & \ddots & \ddots & 0 \\ -\lambda^{\delta-2}\nu^{\delta-2} & 0 & & -\lambda\nu & X+1 \end{vmatrix}$$

Expanding χ_ϵ according to the first row yields

$$\chi_\epsilon = (X - \epsilon a)(X+1)^{\delta-2} + \epsilon Q(X)$$

where $Q(X)$ is a polynomial independent of ϵ, and $a = a_1 + 2a_2\nu + \sum_{i=3}^{\delta} a_i\nu^{i+1}$. Let us now recall a well known result about the sign of the real-part of a polynomial's roots. Let

$$p(X) = p_0X^\delta + p_1X^{\delta-1} + \ldots + p_{\delta-1}X + p_\delta$$

be a polynomial. The Routh table of that polynomial is defined as

$$\begin{array}{ccccc} p_0 & p_2 & p_4 & p_6 & \dots \\ p_1 & p_3 & p_5 & p_7 & \dots \\ r_{3,1} & r_{3,2} & r_{3,3} & \dots & \\ r_{4,1} & r_{4,2} & r_{4,3} & \dots & \\ \vdots & \vdots & \vdots & & \\ r_{\delta+1,1} & & & & \end{array}$$

where the first two rows are given by p, and the other rows $i > 2$ are defined by:

$$(r_{i,1}, r_{i,2}, \ldots) := (r_{i-2,2}, r_{i-2,3}, \ldots) - \frac{r_{i-2,1}}{r_{i-1,1}}(r_{i-1,2}, r_{i-1,3}, \ldots).$$

We use the following result.

Theorem 14 *[Routh-Hurwitz test[42]] A polynomial* $p(X) = p_0X^\delta + p_1X^{\delta-1} + \ldots + p_{\delta-1}X + p_\delta$ *has all its roots with negative real-part if and only if all the* $\delta+1$ *elements of the first column of its Routh table are nonzero and have the same sign.*

Using the Routh-Hurwitz test, we can prove the following result, whose application completes the proof of Lemma 17.

Claim 3 *All the roots of polynomial χ_ϵ have negative real-part for ϵ small enough.*

To establish the claim, let us first consider the family $r_{(i,j)}$ of coefficients of the Routh table of polynomial $(X+1)^{\delta-2}$. It follows from the (direct sense of the) criterion that, since $r_{1,1}=1$ every element of the first column must be positive. Now, if we write

$$\chi_\epsilon(X) = \sum_{j=0}^{\delta-1} b_j X^j = (X-\epsilon A)(X+1)^{\delta-2} + \epsilon Q(X)$$

and call $s_{i,j}$ the family of coefficient of the Routh table of χ_ϵ, we get that:

$$\begin{array}{c} s_{1,j} = b_{\delta-2j+1} = r_{1,j} + \epsilon t_{1,j} \\ s_{2,j} = b_{\delta-2j} = r_{2,j} + \epsilon t_{2,j} \end{array}$$

except for the final coefficient ($s_{2,k}$ if $\delta = 2k$ or $s_{1,k+1}$ if $\delta = 2k+1$), which satisfies

$$b_0 = \epsilon(a + Q(0)).$$

It follows that

$$s_{3,1} = r_{1,2} - \frac{r_{1,1}}{r_{2,1}} r_{2,2} + \epsilon \left(\frac{t_{1,1}}{r_{2,1}} - \frac{r_{1,1} t_{2,1}}{r_{2,1}^2}) r_{2,2} + \frac{r_{1,1}}{r_{2,1}} t_{1,2} \right) + o(\epsilon).$$

Thus, by writing

$$t_{3,1} = (\frac{t_{1,1}}{r_{2,1}} - \frac{r_{1,1} t_{2,1}}{r_{2,1}^2}) r_{2,2} + \frac{r_{1,1}}{r_{2,1}} t_{1,2}$$

we get $s_{3,1} = r_{3,1} + \epsilon t_{3,1} + o(\epsilon)$. Similarly, whenever $r_{i,j}$ is defined, we have

$$s_{i,j} = r_{i,j} + \epsilon t_{i,j} + o(\epsilon)$$

where

$$t_{i,j} = (\frac{t_{i-2,1}}{r_{i-1,1}} - \frac{r_{i-2,1} t_{i-1,1}}{r_{i-1,1}^2}) r_{i-1,j-1} + \frac{r_{i-2,1}}{r_{i-1,1}} t_{i-1,j-1}.$$

Thus, the Routh table of χ_ϵ is as follows:

Case $\delta = 2k$:

$$\begin{array}{ccccc}
r_{1,1} + \epsilon t_{1,1} + o(\epsilon) & r_{1,2} + \epsilon t_{1,2} + o(\epsilon) & \dots & r_{1,k-1} + \epsilon t_{1,k-1} + o(\epsilon) & r_{1,k} + \epsilon t_{1,k} + o(\epsilon) \\
r_{2,1} + \epsilon t_{2,1} + o(\epsilon) & r_{2,2} + +\epsilon t_{2,1} + o(\epsilon) & \dots & r_{2,k-1} + \epsilon t_{2,k-1} + o(\epsilon) & b_0 \\
r_{3,1} + \epsilon t_{3,1} + o(\epsilon) & r_{3,2} + \epsilon t_{3,2} + o(\epsilon) & \dots & r_{3,k-1} + \epsilon t_{3,k-1} + o(\epsilon) & \\
r_{4,1} + \epsilon t_{4,1} + o(\epsilon) & r_{4,2} + \epsilon t_{4,2} + o(\epsilon) & \dots & b_0 & \\
\vdots & \vdots & & & \\
r_{2k-2,1} + \epsilon t_{2k-2,1} + o(\epsilon) & b_0 & & & \\
r_{2k-1,1} + \epsilon t_{2k-1,1} + o(\epsilon) & & & & \\
b_0 & & & &
\end{array}$$

Case $\delta = 2k+1$:

$$\begin{array}{cccccc}
r_{1,1}+\epsilon t_{1,1}+o(\epsilon) & r_{1,2}+\epsilon t_{1,2}+o(\epsilon) & \dots & r_{1,k-1}+\epsilon t_{1,k-1}+o(\epsilon) & r_{1,k}+\epsilon t_{1,k}+o(\epsilon) & b_0 \\
r_{2,1}+\epsilon t_{2,1}+o(\epsilon) & r_{2,2}+\epsilon t_{2,2}+o(\epsilon) & \dots & r_{2,k-1}+\epsilon t_{2,k-1}+o(\epsilon) & r_{2,k}+\epsilon t_{2,k}+o(\epsilon) & \\
r_{3,1}+\epsilon t_{3,1}+o(\epsilon) & r_{3,2}+\epsilon t_{3,2}+o(\epsilon) & \dots & r_{3,k-1}+\epsilon t_{3,k-1}+o(\epsilon) & b_0 & \\
\vdots & \vdots & & & & \\
r_{2k-1,1}+\epsilon t_{2k-1,1}+o(\epsilon) & b_0 & & & & \\
r_{2k,1}+\epsilon t_{2k,1}+o(\epsilon) & & & & & \\
b_0 & & & & &
\end{array}$$

Since for all $i, r_{i,1}$ is positive, it follows that all the elements of the first column but the last are positive for ϵ small enough. It just remain to prove that $b_0 > 0$. For that purpose, we establish the following result:

$$|J_\epsilon| = (-1)^\delta P'_\epsilon(\nu). \tag{5.5}$$

Before proving that equality, let us complete the proof of the claim. Since χ_ϵ is the characteristic polynomial of J_ϵ, it's last coefficient is $b_0 = (-1)^{\delta-1}|J_\epsilon| = -P'_\epsilon(\nu) > 0$, from Eq. 5.5. Thus all the elements of the first column of the Routh table of χ_ϵ are positive and the equilibrium is stable by (the indirect sense of) the Routh Hurwitz Criterion. This completes the proof.

Now, it just remain to prove Eq. 5.5. Let A_k be the following $k*k$ matrix:

$$A_k = \begin{vmatrix}
\epsilon(a_1 + 2a_2\nu + \sum_{i=3}^n a_i\nu^{i+1}) & \epsilon\frac{a_3\nu}{\lambda} & & \dots & \epsilon\frac{a_{k+1}\nu}{\lambda^{k-1}} \\
2\lambda\nu & -1 & & 0 & \\
\lambda^2\nu^2 & \lambda\nu & -1 & & 0 \\
\vdots & 0 & \ddots & \ddots & 0 \\
\lambda^{k-1}\nu^{k-1} & 0 & & \lambda\nu & -1
\end{vmatrix}.$$

By this definition, we get

$$A_{\delta-1} = |J_\epsilon|.$$

Expanding A_k along the last row yields

$$A_k = -A_{k-1} + (-1)^{k-1}\epsilon\frac{a_{k+1}\nu}{\lambda^{k-1}}B_{k-1}$$

where

$$B_j = \begin{vmatrix}
2\lambda\nu & -1 & & & \\
\lambda^2\nu^2 & \lambda\nu & -1 & & 0 \\
\vdots & & \ddots & \ddots & \\
\lambda^{j-1}\nu^{j-1} & 0 & & \lambda\nu & -1 \\
\lambda^j\nu^j & & & & \lambda\nu
\end{vmatrix}.$$

Expanding B_j yields

$$Bj = \lambda\nu B_{j-1} + \lambda^j \nu^j.$$

Furthermore,

$$B_2 = \begin{vmatrix} 2\lambda\nu & -1 \\ \lambda^2\nu^2 & \lambda\nu \end{vmatrix} = 3\lambda^2\nu^2$$

Thus, by induction,

$$B_j = (j+1)\lambda^j\nu^j$$

from which it follows that

$$A_k = -A_{k-1} + (-1)^{k-1} k\epsilon a_{k+1}\nu^k.$$

Furthermore,

$$A_2 = \begin{vmatrix} \epsilon(a_1 + 2a_2\nu + \sum_{i=3}^{\delta} a_i\nu^{i+1}) & \epsilon\frac{a_3\nu}{\lambda} \\ 2\lambda\nu & -1 \end{vmatrix} = -\epsilon(a_1 + 2a_2\nu + 3a_3\nu^3 + \sum_{k=4}^{\delta} a_i\nu^{i+1})$$

Thus, by induction,

$$\begin{aligned} A_k &= (-1)^{k-2}\left(A_2 + \textstyle\sum_{i=3}^{k} \epsilon i a_{i+1}\nu^i\right) \\ &= (-1)^{k-1}\epsilon\left(\textstyle\sum_{i=1}^{k+1} i a_i \nu^{i-1} + \sum_{i=k+2}^{\delta} a_i\nu^{i-1}\right). \end{aligned}$$

Finally, we get:

$$|J_\epsilon| = A_{\delta-1} = (-1)^\delta P'_\epsilon(\nu),$$

which completes the proof of Eq. 5.5. □

5.3 Conclusion

In this chapter, we have given a complete characterization of the numbers that can be computed using *Large-Population Protocols*. We showed that these numbers are exactly the algebraic real numbers in $[0, 1]$. In Lemma 15, we also showed that considering LPPs in which the transition rules a probabilistic with rational probabilities did not increase the computational power. The reasoning in Lemma 15 would also hold if randomized probabilities were replaced with any computable probability values. This result indicates that the uniformly random scheduler for interaction is already a very powerful source of randomness that can be used to simulate any random oracle with computable probabilities.

However, the definition of computation given in chapter 4 requires an execution to start sufficiently close to the number ν being computed. We prove the existence of a neighbourhood for any algebraic number ν that would guarantee eventual convergence of a protocol computing ν. We do not, however, give a bound on the size of a security ball around ν. In other words, we do not know how close to ν the protocol has to start to be able to eventually compute ν.

Also not addressed is the matter of multiple possible equilibria for a single protocol. Indeed a LPP may admit multiple stable equilibria and, therefore, compute several distinct numbers depending on where the protocol starts.

Conclusion and Perspectives

Conclusion

In this dissertation we have presented several results extending our understanding of population dynamics in general, and of population protocols in particular. A central result on population protocols, due to Angluin *et al.* [3], is that the predicates computable by a population protocol are exactly those definable in Presburger arithmetic. In Part I, we have established how a population protocols can be derived from any two-player game in normal form, by having a population of agents play the game repeatedly against one another in pairwise encounters and update their strategy by following the *PAVLOV* behaviour. We called *Pavlovian* a population protocol that can be obtained from two-player games in this manner. We considered the computational power of such Pavlovian protocols. We proved that restricting population protocols to Pavlovian protocols does not, in fact, reduce the computational power of the model, as far as computable predicates are concerned. In particular, we showed how, given a semilinear predicate ψ, we can construct a series of games such that running the associated Pavlovian protocols in parallel, we call this a multi-protocol, computes ψ.

We then studied the case of symmetric games. We showed that any given population protocol can be simulated by a symmetric population protocol. This result, however, does not extend to Pavlovian protocols. Indeed, we proved that no symmetric Pavlovian protocol could even detect accurately if three or more occurrences of an input symbol were present in the input configuration. We introduced *exclusive* Pavlovian protocol, in which a dissatisfied agent changes his strategy even if his current strategy is already optimal against his opponent's. We showed that exclusive Pavlovian protocols are more powerful than symmetric Pavlovian protocols, being able to detect if 3 or more occurrences of an input symbol are present in the population. It was unclear however exactly how powerful they were.

In Part II, we studied an extension of population protocols, called Large Population Protocols, designed to capture the behaviour of very large population, observing how the fraction of the population in each different state evolves. We approximated such large populations by a virtual infinite population whose dynamic corresponds to the solution of a particular ordinary differential equation which can be derived from the protocol's transition rules. We showed how this virtual population is indeed the limit behaviour of a population whose size grows to infinity. We then defined computations with Large Population Protocols to be the convergence to a stable equilibrium this infinite population when times goes to infinity. In

other words, computation by a LPP means a double convergence in time and population size towards an equilibrium. Indeed, we showed that if a LPP computes a real number $\nu \in [0, 1]$, then ν can be approximated within a margin of ϵ with probability $1 - \epsilon$ using a population of size n and after a time t, where both n and t are inversely polynomial in ϵ, for any $\epsilon > 0$ if one guarantees that the protocol starts within a neighbourhood of the equilibrium associated with ν.

Finally, in chapter 4, we determined which real numbers can be computed by a LPP. We proved that no transcendent numbers can be computed by a LPP as a consequence of Tarski's effective procedure for quantifier elimination over real-closed fields. Reciprocally, we constructively showed how to design a protocol to compute any algebraic real number in $[0, 1]$, thereby completely characterizing the computational power of large population protocols.

Perspectives

In the spirit of this dissertation, some questions remain unanswered an new questions have arisen that deserve investigation. As mentioned before, it remains unclear which predicates can be computed by symmetric and exclusive Pavlovian protocols. It appears that symmetric Pavlovian protocols are quite weak, being unable to even detect three occurrences of an input symbol. It is unclear, however, if counting up to two is the only thing they can do. Such a clear result, similar to that achieved by Angluin *et al.* in the case of delayed observation protocols [5] would be worth investigating. Similarly, it remains to be determined what exclusive Pavlovian protocols are able to do. The protocol constructed to count up to 2^k in chapter 3 relies heavily on the pairwise structure of the interactions. It does not seem that it can be readily adapted to even other simple counting predicates. We do not, however, have a proof that nothing more can be computed.

On the topic of large population protocols, we do have a complete characterisation of what real numbers can be computed. We even give the size of the population needed, and the time required to compute any given number within a given margin of error of $\epsilon > 0$ with high probability. For such a convergence to be possible, however, we require the computation to start within an open ball around the desired equilibrium. The toy example computing $\frac{\sqrt{2}}{2}$ shows that in some cases the convergence can be insured regardless of the initial configuration. The existence of conjugated pairs of algebraic numbers in $[0, 1]$ (numbers that share the same minimal polynomial) indicates that it is unlikely that this is possible in general. It is clear that some protocols may accept multiple stable equilibria, in which case convergence to a given root is only possible if starting within a strict part of the simplex of possible configurations, and that some trajectories would be never-converging. We do not know a general expression of the security radius guaranteeing eventual convergence of a large population protocol towards a specific value, neither do we know the phase portrait of possible trajectories corresponding to a LPP. Additionally, we showed on the toy example, that at least in some specific cases, we could be more precise about the approximation made by comparing a finite population of size n to the virtual infinite population. We showed that, for the toy example, the difference between the two was of order $\frac{1}{\sqrt{n}}$. The method used

in this example does not scale well to general population protocols and it remains to be seen if similar results could be achieved using a different approach for the general case.

Several of the questions raised about population protocols should also be answered for Pavlovian protocols and LPPs. For instance, can the fault-tolerance properties proved by Delporte-Gallet *et al.* [23] be extended to Pavlovian protocols ? Similarly, it is unclear what the impact of restricting the interaction graph of a Pavlovian protocol would be. The community protocol allows for some level of tolerance towards Byzantine agents and could be used to model a small amount of memory for players able to remember a finite number of their past adversaries. LPPs, it seems, would be largely impacted by a fixed fraction of Byzantine agents, while a finite number of Byzantine agents would be unnoticeable in an infinite population.

It is also unclear if replacing the uniform random scheduler with some other scheduler could impact computational power. It is however possible that, similarly to simulating randomised transition as is shown in chapter 5, such a scheduler could be simulated with the universally random scheduler under certain conditions.

Finally, going back to phenomena occurring in nature, dynamic models can be found in biology. Ants, for example, have been observed to use pairwise communication to develop some form of error correction: when out gathering food, a small fraction of ants would sometime be confused and go in the wrong direction (bringing food back to the food source, for example) and, by interacting with ants going in the other direction, the ants with the wrong opinion would generally be convinced to go in the right direction, while sometimes, if too many confused ants were in the same area they might convince other ants of the wrong idea. It would be interesting to study if such a natural consensus process could be expressed in terms of a large population protocol ? Such questions are currently under investigation in collaboration with Ofer Feinerman and Amos Korman.

Bibliography

[1] Dana Angluin, James Aspnes, Melody Chan, Michael J. Fischer, Hong Jiang, and René Peralta. Stably computable properties of network graphs. In *First IEEE Conference on Distributed Computing in Sensor Systems (DCOSS)*, volume 3560 of *LNCS*, pages 63–74. Springer-Verlag, 2005.

[2] Dana Angluin, James Aspnes, Zoë Diamadi, Michael J. Fischer, and René Peralta. Computation in networks of passively mobile finite-state sensors. In *23rd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 290–299, 2004.

[3] Dana Angluin, James Aspnes, and David Eisenstat. Stably computable predicates are semilinear. In *25th ACM symposium on Principles of Distributed Computing (PODC)*, pages 292–299, 2006.

[4] Dana Angluin, James Aspnes, and David Eisenstat. Fast computation by population protocols with a leader. *Distributed Computing*, 21(3):183–199, September 2008.

[5] Dana Angluin, James Aspnes, David Eisenstat, and Eric Ruppert. The computational power of population protocols. *Distributed Computing*, 20(4):279–304, 2007.

[6] Dana Angluin, James Aspnes, Michael J. Fischer, and Hong Jiang. Self-stabilizing population protocols. *TAAS*, 3(4), 2008.

[7] James Aspnes and Eric Ruppert. An introduction to population protocols. *Bulletin of the EATCS*, 93:106–125, 2007.

[8] Guillaume Aupy and Olivier Bournez. On the number of binary-minded individuals required to compute $\sqrt{\frac{1}{2}}$. *Theoretical Computer Science*, 412(22):2219–2456, 2010.

[9] Robert Axelrod. *The Evolution of Cooperation.* Basic Books, 11 edition, 1984.

[10] Joffroy Beauquier, Janna Burman, Julien Clément, and Shay Kutten. On utilizing speed in networks of mobile agents. In Andréa W. Richa and Rachid Guerraoui, editors, *29th ACM symposium on Principles of Distributed Computing (PODC)*, pages 305–314. ACM, 2010.

[11] Joffroy Beauquier, Janna Burman, and Shay Kutten. A self-stabilizing transformer for population protocols with covering. *Theoretical Computer Science*, 412(33):4247–4259, 2011.

[12] Joffroy Beauquier, Julien Clément, Stéphane Messika, Laurent Rosaz, and Brigitte Rozoy. Self-stabilizing counting in mobile sensor networks with a base station. In Andrzej Pelc, editor, *DISC*, volume 4731 of *L*, pages 63–76. Springer, 2007.

[13] Ken Binmore. *Fun and games*. Heath, 1992.

[14] Olivier Bournez, Jérémie Chalopin, Johanne Cohen, Xavier Koegler, and Mikaël Rabie. Computing with pavlovian populations. In Antonio Fernández Anta, Giuseppe Lipari, and Matthieu Roy, editors, *15th International Conference On Principles Of Distributed Systems (OPODIS)*, volume 7109 of *LNCS*, pages 409–420. Springer, 2011.

[15] Olivier Bournez, Philippe Chassaing, Johanne Cohen, Lucas Gerin, and Xavier Koegler. On the convergence of population protocols when population goes to infinity. *Applied Mathematics and Computation*, 215(4):1340–1350, 2009.

[16] Olivier Bournez, Pierre Fraigniaud, and Xavier Koegler. Comuting with large populations using interactions. *to appear in MFCS 2012*, Bratislava, Slovakia, August 27 - 31, 2012.

[17] George W. Brown. Iterative solution of games by fictitious play. In *Activity Analysis of Production and Allocation*, Cowles Commission Monograph No. 13, pages 374–376. John Wiley & Sons Inc., New York, N. Y., 1951.

[18] Augustin Chaintreau, Pan Hui, Jon Crowcroft, Christophe Diot, Richard Gass, and James Scott. Impact of human mobility on opportunistic forwarding algorithms. *IEEE Transactions on Mobile Computing*, 6(6):606–620, 2007.

[19] Chen Chung Chang and H. Jerome Keisler. *Model theory*. North Holland, 1990.

[20] Ioannis Chatzigiannakis, Othon Michail, Stavros Nikolaou, Andreas Pavlogiannis, and Paul G. Spirakis. Passively mobile communicating machines that use restricted space. In *Proceedings of the 7th ACM ACM SIGACT/SIGMOBILE International Workshop on Foundations of Mobile Computing*, FOMC '11, pages 6–15, New York, NY, USA, 2011. ACM.

[21] Ioannis Chatzigiannakis and Paul G. Spirakis. The dynamics of probabilistic population protocols. In *22nd International Symposium on Distributed Computing (DISC)*, volume 5218 of *LNCS*, pages 498–499, 2008.

[22] Francis Comets and Thierry Meyre. *Calcul stochastique et modèles de diffusions*. Dunod Paris, 2006.

[23] Carole Delporte-Gallet, Hugues Fauconnier, Rachid Guerraoui, and Eric Ruppert. When birds die: Making population protocols fault-tolerant. In *2nd IEEE Conference on Distributed Computing in Sensor Systems (DCOSS)*, volume 4026 of *LNCS*, pages 51–66. Springer, 2006.

[24] Jean-Pierre Demailly. *Analyse Numérique et Equations Différentielles*. Presses Universitaires de Grenoble, 1991.

[25] Martin Dyer, Leslie Ann Goldberg, Catherine Greenhill, Gabriel Istrate, and Mark Jerrum. Convergence of the iterated prisoner's dilemma game. *Combinatorics, Probability and Computing*, 11(2):135–147, March 2002.

[26] A. Fernández, V. Gramoli, E. Jiménez, A.-M. Kermarrec, and M. Raynal. Distributed slicing in dynamic systems. In *27th IEEE Int. Conference on Distributed Computing Systems (ICDCS)*, 2007.

[27] Michael J. Fischer and Hong Jiang. Self-stabilizing leader election in networks of finite-state anonymous agents. In *10th International Conference On Principles Of Distributed Systems (OPODIS)*, pages 395–409, 2006.

[28] Laurent Fribourg, Stéphane Messika, and Claudine Picaronny. Coupling and self-stabilization. *Distributed Computing*, 18(3):221–232, February 2006.

[29] Drew Fudenberg and David K. Levine. *The Theory of Learning in Games*, volume 1 of *MIT Press Books*. The MIT Press, 1998.

[30] V. Gramoli, Y. Vigfusson, K. Birman, A.-M. Kermarrec, and R. van Renesse. Slicing distributed systems. *IEEE Trans. Computers*, 58(11):1444–1455, 2009.

[31] Rachid Guerraoui and Eric Ruppert. Names trump malice: Tiny mobile agents can tolerate byzantine failures. In *Proceedings of the 36th Internatilonal Collogquium on Automata, Languages and Programming: Part II*, ICALP '09, pages 484–495, Berlin, Heidelberg, 2009. Springer-Verlag.

[32] Morris W. Hirsch, Stephen Smale, and Robert Devaney. *Differential Equations, Dynamical Systems, and an Introduction to Chaos*. Elsevier Academic Press, 2003.

[33] J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, 1998.

[34] Josef Hofbauer and Karl Sigmund. Evolutionary game dynamics. *Bulletin of the American Mathematical Society*, 4:479–519, 2003.

[35] Pan Hui, Augustin Chaintreau, James Scott, Richard Gass, Jon Crowcroft, and Christophe Diot. Pocket switched networks and human mobility in conference environments. In *ACM SIGCOMM workshop on Delay-tolerant networking (WDTN)*, pages 244–251, 2005.

[36] M. Jelasity and A.-M. Kermarrec. Ordered slicing of very large-scale overlay networks. In *6th IEEE Int. Conference on Peer-to-Peer Computing (P2P)*, pages 117–124, 2006.

[37] Philo Juang, Hidekazu Oki, Yong Wang, Margaret Martonosi, Li Shiuan Peh, and Daniel Rubenstein. Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. In *10th ACM conference on Architectural support for programming languages and operating systems (ASPLOS)*, pages 96–107, 2002.

[38] David P. Kraines and Vivian Y. Kraines. Pavlov and the prisoner's dilemma. *Theory and Decision 26, 47–79.*, 1989.

[39] Paul Langevin. Sur la théorie du mouvement brownien. *Compte Rendu de l'Académie des Sciences*, 146:530–533, 1908.

[40] A. J. Lotka. Analytical Note on Certain Rhythmic Relations in Organic Systems. *Proceedings of the National Academy of Sciences of the United States of America*, 6(7):410–415, July 1920.

[41] Alfred J. Lotka. Contribution to the Theory of Periodic Reactions. *The Journal of Physical Chemistry*, 14(3):271–274, January 1910.

[42] Gjerrit Meinsma. Elementary proof of the routh-hurwitz test. *Systems and Control Letters*, 25(4):237 – 242, 1995.

[43] Marvin L. Minsky. *Computation: finite and infinite machines.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1967.

[44] Michael Mitzenmacher and Eli Upfal. *Probability and Computing. Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, 2005.

[45] James Dickson Murray. *Mathematical Biology. I: An Introduction.* Springer, third edition, 2002.

[46] Martin Nowak and Karl Sigmund. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, 364(6432):56–58, 1993.

[47] Mojzesz Presburger. Uber die Vollstandig-keit eines gewissen systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In *Comptes-rendus du 1er Congrès des Mathematiciens des Pays Slaves*, pages 92–101, 1929.

[48] Dandiel W. Stroock and S. R. Srinivasa Varadhan. *Multidimensional Diffusion Processes.* Springer, 1979.

[49] Jörgen W. Weibull. *Evolutionary Game Theory.* The MIT Press, 1995.