Universitat de Barcelona

An agent-based model for Lewis signalling games

Final essay in *Philosophy of language and cognition*

Marcel Ruland

hand-in date: January 23, 2019
Word count: 5447

Instructor: Manolo Martínez

Contents

1 Introduction

Lewis signalling games (henceforth LSG), first introduced by Lewis (1969), are abstract games which allow for a signalling language to eventually emerge from initially random signals. Following Barrett (2007, p. 530 ff.), an LSGworks as follows: There is a set of states of the world S, a set of signals (or terms) T, and a set of acts A. There is a mapping from acts to states of the world, such that every act corresponds to one state of the world.[1] There is a *sender*, who can observe the current state of the world and a *receiver*, who cannot observe the current state of the world. In each round exactly one state of the world $s \in S$ holds, i.e. is the current state of the world. The sender will observe the current state of the world and then choose a signal $t \in T$ and send it to the receiver. The receiver will observe the signal sent by the sender and then choose an act $a \in A$ to perform. An iteration is won if the act a matches the current state of the world s and lost if it does not. Both the sender and the receiver know whether the previous iteration was a success or a failure and adapt their strategies for choosing signals and acts according to some learning function. Commonly, states of the world are distributed uniformly and sender and receiver start out with randomly choosing a signal and an act respectively, but this is not a formal requirement for the game to count as an LSG.

A note on terminology: For the scope of this essay, I refer to one cycle of new-state-of-the-world → sender-chooses-signal → receiver-chooses-action → communication-succeeds-or-fails as an *iteration*. Many iterations one after the other up until some stopping condition is met is referred to as a *game*. Many games played under the same conditions one after the other, which may then be used to measure some average metric, are referred to as a *run*.

Agent-based modelling (henceforth ABM[2]) is a fairly recent modelling paradigm that has its roots in cellular automata and complexity theory (Heath and Hill, 2014). The general conceptual thought behind ABM is to model the behaviour of entities

[1] Barrett (2007) leaves unclear if that mapping necessarily has to be bijective (i.e. for every a there is exactly one corresponding s and for every s there is at least one corresponding a), but every signalling game involved in this essay fulfils this condition.

[2] I use the abbreviation ABM to refer either to one specific agent-based model or to the modelling paradigm of agent-based modelling as a whole. However, it always becomes clear from context, which of the two terms is meant.

in an environment. Following Grimm and Railsback (2005); Railsback and Grimm (2011) in an ABM, there are individual *agents* representing individual entities of one or several kinds (such as for instance cars, sheep, or viruses). These entities interact with each other, with an *environment*, or (typically) both. The environment itself has characteristics which influence the agents. It can play a major role in the simulation (such as a forest providing food, shelter, and other things to a population of animals) or be virtually inexistent and reduced to a simple integer variable (as is the case in the present essay, see section 2). Typical applications include (but are by far not limited to) the simulation of traffic flow in a road network, spreading of a virus in a population of organisms, and changes in real estate prices in a city. The strength of ABM, in comparison to other modelling paradigms, is that by modelling each individual agent, one can observe how the sum of individual actions of agents give rise to phenomena which were not explicitly programmed into the model (such as the creation of ant corridors, which are not intentionally created by any individual ant, but are a byproduct of the fact that ants directly follow each other in straight lines, Wilensky (1997)), so-called emergent behaviour.

This essay aims to combine both paradigms. An ABM modelling a LSG will be created where there is a population of senders and a population of receivers. Initially the number of signals available will be less than the number of states of the world and actions, but will be increased with time. The aim is to determine how delaying the introduction of signals into the game influences the speed of convergence towards *perfect communication*. Perfect communication in an LSG is achieved if "each state of the world corresponds to a term in the language and each term corresponds to an act that matches the state of the world, so each signal leads to a successful action" (Barrett, 2007, p. 530, there referred to as "perfect Lewis signalling system"). The model, described in detail in section 2, has quite some functionality for choosing parameters which will not be used in this essay, for simple reasons of scope.

2 Model description

Replication of previous results is one of the core foundations of empirical science. Yet one of the main challenges of ABM remains to be replicability. The smallest implementational details, which may elude the programmers' attention during the entire modelling process, can lead to emergent behaviour or just as well prevent it from forming. It is therefore essential that models be described in such a way that they can be replicated in great detail and their results be reproduced successfully. For this purpose, Grimm et al. (2006, 2010) have developed the *Overview, Design concepts, and Details* protocol (henceforth ODD), whose purpose is to establish a standard for describing ABMs in the scientific literature across disciplines and which has received several extensions since (e.g. Müller et al., 2013, for models involving human decision making).

In my description of the model, I follow the ODD which is subdivided into the following subsections, which I describe one after the other: 1. Purpose 2. Entities, state variables, and scales 3. Process overview and scheduling 4. Design concepts 5. Initialisation I acknowledge that for a relatively simple model as the one described here, this exploiting the entire ODD standard may be a bit over the top. Yet I prefer being too explicit to being too vague. The model has been implemented in Netlogo (Wilensky, 1999) and the entire source code is available at `https://github.com/prestoaghitato/plc_lsg/blob/master/netlogo/plc_lsg.nlogo`

2.1 Purpose

The model is run to understand how LSGs are influenced by various parameters such as number of signals, number of states of the world, number of actions, or population size of senders and receivers. The main goal during implementation was to look at how convergence speed towards perfect communication is influenced by delaying the introduction of signals into an ongoing game, but the model can easily be used to investigate other parameters. Possible extensions include implementing different learning functions or making the number of possible states of the world and the number of available actions separate parameters.

2.2 Entities, state variables, and scales

There are two main kinds of entities, `senders` and `receivers`, the populations of which are equal in size. To be precise, there is a third kind of entity involved in the model: the link. However, the purpose of links differs drastically from that of senders and receivers and their description is therefore held separately from that of senders and receivers. The `senders` have two state variables: `urns` and `chosen-signal`. `urns` is a list of integer lists for implementing the urns in an urn learning function, see paragraph on learning in subsection 2.4. Every nested list represents one urn, the integers within it represent balls to be drawn from it. `chosen-signal` is an integer corresponding to the signal the sender will send in the current iteration.

`receivers` are structured almost identically. The urns are also lists of integer lists, but are structured differently (see again paragraph on learning in subsection 2.4). Instead of a `chosen-signal` variable, they have a `chosen-action` variable. In addition, they also have a `received-signal` variable, which stores the signal received from a sender.

Senders and receivers are connected by `links`, which carry a signal in their `signal` variable, their sole state variable.[3] This `signal` variable can be read only by the sender and the receiver connected to the link. The environment is simply a state of the world, implemented as a global integer variable, and nothing else. The following parameters of the model can be modified:

- number of possible states of the world (the number of possible actions is always equal to this value)—`num-world-states`
- number of available signals—`num-signals`
- population size (equal for senders and receivers)—`population-size`
- number of balls added in case of success—`num-add-balls`
- number of balls removed in case of failure—`num-remove-balls`

Furthermore, an interval *n* can be given such that after *n* iterations a new signal is introduced into the ongoing game (parameter `signals-interval`). This mechanism is optional and can be switched off entirely.

[3] Strictly speaking, a link also stores information on which two agents it connects in its `end1` and `end2` variables. In Netlogo, links are a primitive kind of agent so these connections were not explicitly programmed into the links when writing the model. Agents in Netlogo have many other default state-variables which play no role whatsoever in this model.

2.3 Process overview and scheduling

Every iteration begins with randomly choosing a state of the world and creating a randomised one-to-one mapping from senders to receivers. Links are then created and connected to senders and receivers according to this random mapping. Senders have access to the state of the world and choose a signal based on it, which they then pass on to their respective link. Once the links carry the sent signal, receivers have access to it and read it. They choose an action based on the received signal and check whether the action matches the state of the world. Receivers then pass on an arbitrary, conventional value (here 42) on to the links, which the receivers can read and interpret as *success*. Lastly, senders and receivers run a learning function according to whether the last iteration was a success or a failure.

The careful reader will notice that the previous paragraph seems to contradict the above given definition for an LSG. Receivers cannot observe the state of the world, yet in this model description it is the receivers who check whether their performed action corresponds to the current state of the world. While this may seem a violation of the rules of an LSG, it is in fact not. Once signals and actions have been chosen, the simplest way of implementing the passing on of the correct success/failure information to the correct senders/receivers just happened to be having the receivers themselves check the global state of the world variable and then informing their corresponding senders via the still existent links. At no other time do the receivers access the state of the world variable, thus not violating the theoretical definition of an LSG.

2.4 Design concepts

Emergence The number one (and only) emergent behaviour predicted here is that of perfect communication. From an initially random distribution of chosen signals and chosen actions emerges (ideally) a perfect one-to-one mapping from states of the world to signals, from signals to actions, and from actions to states of the world, such that every performed act corresponds to the current state of the world.

Adaptation Senders have as many urns as there are possible states of the world. Receivers have as many urns as there are available signals. A sender chooses a signal by randomly drawing a ball from the urn corresponding to the current state of the world. A receiver chooses an action by randomly drawing a ball from the urn corresponding to the received signal. Initially, every sender's urn contains exactly one ball corresponding to every possible state of the world and every receiver's urn contains exactly one ball corresponding to every available action.

Objectives Senders and receivers aim to maximise the percentage of iterations in which communication is successful, i.e. in which the current state of the world matches the action chosen by the receiver. They do so by running a learning function after every iteration, described in the following paragraph.

Learning The implemented learning function is the urn learning function as described in Barrett (2006, sec. 2, unpaginated preprint). Senders start off with one ball corresponding to every possible signal in each of their urns and as many urns as there are possible states of the world. In a game with three possible states of the world and two available signals, the initial configuration will look like this: `[[0 1] [0 1] [0 1]]` For senders, each of their urns corresponds to one state of the world (bijective mapping) and they will always consult the urn corresponding to the current state of the world.

Receivers start off with one ball corresponding to every possible action in each of their urns and as many urns as there are signals available to the senders. In the same game (three possible states of the world, two available signals), their initial urn configuration looks as follows: `[[1 2 3] [1 2 3]]` For receivers, each of their urns corresponds to one available signal (bijective mapping) and they will always consult the urn corresponding to the received signal.

In case of success, senders and receivers will add more balls with the chosen signal/action to the consulted urn. In case of failure, they will remove balls with the chosen signal/action, but *only* under the condition that after having removed the balls, there remains at least one ball with the given signal/action so that no signal or action can be removed from the game entirely. If, for example, a sender has four balls with signal 2 in the consulted urn and is required by its learning function to remove five balls of that kind, he will only remove three so that one ball with signal 2 remains in the urn. The number of added or removed balls is determined by the value of two parameters.

Prediction Agents cannot predict the future values of any variables.

Stochasticity The state of the world of a round is determined by randomly drawing a value from a uniform distribution. Furthermore, within each of the steps described in subsection 2.3, the order in which senders and receivers perform their actions is also randomised.

Observation A game is considered to have converged to perfect communication if *the moving average (mean) of communication quality over the past 1000 rounds is > 0.8 for 100 consecutive rounds.* Communication quality here refers to the percentage of sender-receiver pairs that have managed to communicate successfully in an iteration. If this criterion is not met within one million rounds, then the game is considered to have failed to converge. The 0.8 ratio threshold is in line with Barrett (2006, 2007, p. 533). In Barrett (2006, sec. 2, unpaginated preprint), the author states that "[a]fter 10^6 plays the ratio of successful actions to the number of plays, the signal success rate, is typically better than 0.999", but is not precise in how exactly this is measured (see subsection 3.3). In trial runs, the condition of 100 consecutive rounds has proven to be a reliable indicator for convergence and has therefore been chosen.

2.5 Initialisation

The model is initialised by creating a population of senders and receivers with urns in initial configuration (see paragraph on learning in subsection 2.4), setting the number of available actions equal to the number of possible states of the world, and resetting the counter keeping track of the number of iterations played.

3 Results

The results section is subdivided into three main parts. The first two parts focus each on one of two obvious questions the model brings with it. A parameter not included in traditional LSGs is that of population size. Having more than one speaker or receiver appears to be a novelty and therefore naturally invites investigation. The influence of this new parameter on convergence speed is the object of interest for subsection 3.1. The second intuitive object of inquiry is the introduction of new signals as the iterations of a game progress. Subsection 3.2 deals with this parameter. The third subsection is a little different in nature. Here, I aim to reproduce some of the empirical results of Barrett (2006). Replication of the results is not successful in all cases, although a number of core findings hold in both data sets.

3.1 Population size

In a first experiment, to investigate the effect of population size on convergence speed, the model was run with the following parameter settings:

```
num-signals = 10

num-remove-balls = 1

num-add-balls = 5

num-world-states = 10
```

`population-size` was incremented step-wise from 1, 2, 3, ..., 98, 99, 100 and ten games were performed for every value.[4]

The value chosen for `num-add-balls` may seem unusually high, but this is motivated as follows. Small populations tend to not converge if the value for this parameter is too high (close to 10), whereas large populations tend to not converge if it is too low (close to 1). Non-convergent games are of little interest when one's aim is to investigate convergence speed. Assigning a value of 5 to `num-add-balls` has proven to reliably lead to convergence with both smaller and larger populations. In fact, out of the one thousand games run for this experiment, only four games did not

[4] 10 games are, admittedly, less than one would have liked given the extent of variation shown in figure 1; unfortunately the complexity of running the model with high values for population-size was such that performing e.g. 100 games for every value were just not feasible.

converge within one million iterations. The respective population sizes of those four games were 3, 4, 12, and 48.

The predicted result for this simulation is that with increasing population size the average number of iterations until convergence will also increase. If a given sender and a given receiver are paired in the random mapping and communicate successfully, then they will adapt their strategies only to each other, but not to the rest of the population. This is quite obviously not an issue if population size is equal to one, because the same sender and receiver will be paired over and over again. But as population size increases, the likelihood of being paired with the same sender or receiver decreases and therefore iterations are needed not only for convention to emerge within a single pair but within many pairs and eventually across an entire population.

Figure 1 shows the mean and median convergence speeds for all values of `population-size` respectively. One can see that the mean values consistently are always at least as high, but usually higher, than their median counterparts. This indicates rare runs with very late convergence, increasing the means more than the medians.[5] Visually, the plot seems to affirm the predicted statistical relationship between the two variables and indeed there appears to be a positive correlation of $R^2 = 0.55$. One would expect this relation to be logarithmic, because for every pair added to the population the population grows by a factor of $\frac{1}{n+1}$ (n being the population size), but the simulation is not nearly extensive enough to affirm or deny this hypothesis.

3.2 Introducing signals into an ongoing LSG

The simulations in this subsection are concerned with the effects of delaying the introduction of signals into the game. This has been done on both a macro and a micro scale. A look at the results of the first, larger scale (which was consequently dubbed macro scale) has shown that the intervals chosen for the parameter `signals-interval` were simply much too high to result in producing any interesting effects whatsoever.

[5] All other experiments have shown similar characteristics with rare and high outliers. I have therefore decided to only consider medians as a measure of centrality from now on, which are much less affected by outliers as compared to means.

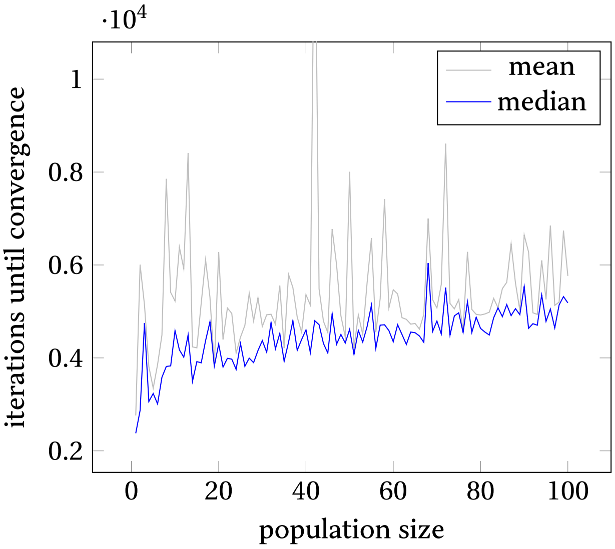

Figure 1: Effect of population size on convergence speed; x-axis indicates population size, y-axis indicates mean and median number of iterations until convergence to perfect communication; the outlier not in view for a population size of 42 has a value of 14450; R^2 = 0.55 for median values.

Macro scale

In the first run, the model was initialised with the following parameter settings:

```
population-size = 10

num-world-states = 10

num-signals = 1

num-remove-balls = 1

num-add-balls = 3
```

More signals were then introduced after a certain amount of iterations—given by the parameter `signals-interval`—had passed. The values tested for `signals-interval` were 100, 200, 300, ..., 4000, 4100, 4200. For every value of the parameter, one hundred games were performed.

Figure 2 shows a somewhat dull linear relationship (R^2 = 0.97). This is, in hindsight, no surprise as figure 1 shows a convergence median of 4585 games for a population of size 10. Introducing signals into the game long after the 4585th round has passed is bound to slow convergence down. The linear relationship one can see here is

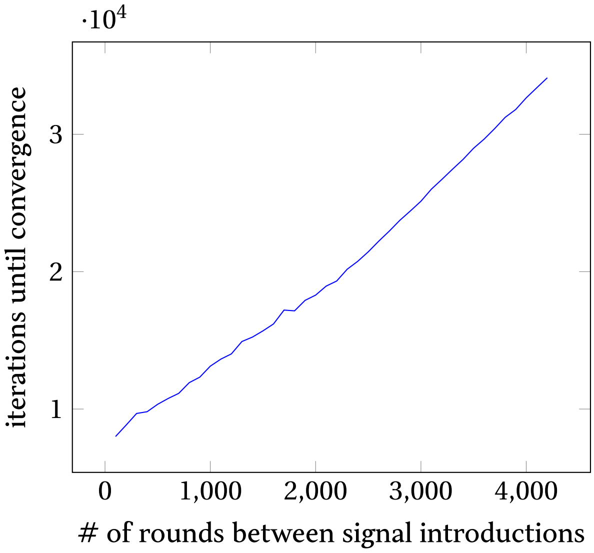

Figure 2: Effect of delayed introduction of signals on convergence speed; x-axis indicates the number of iterations that passed before another signal was introduced (value of `signals-interval`; y-axis indicates the median number of iterations until convergence; $R^2 = 0.97$

simply representing the linear fashion in which the value of `signals-interval` has been increased. A truly interesting result would be a later introduction of signals that speeds up median convergence speed, instead of slowing it down. In order to potentially produce such results it is necessary to zoom in and choose much smaller values for `signals-interval`.

Nevertheless these first simulations may show to provide some interesting data. Figure 3 shows how communication quality evolved along on of the games played as part of the second experiment, and brings much less expected results to light. In the game displayed, a new signal was introduced every 4200 rounds. One can see that communication quality remains close to 0.1 without significant variation up until around the 25000th iteration. Around this iteration, communication quality experiences a sudden increase. This is strikingly different from games that start out with all signals available from the beginning. In those canonical games typically, communication quality typically also increases in a sigmoid pattern[6], but not nearly

[6] I.e. starting with a low increase, which then gets higher and then lower again, resulting in an *S*-like shape as visible in the figure.

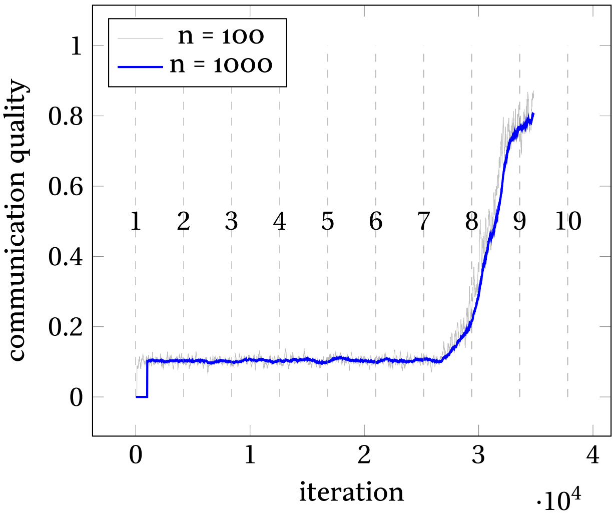

Figure 3: Communication quality in a game where new signals were introduced with a frequency of 1 signal per 4200 rounds played; x-axis indicates iterations, y axis indicates communication quality; grey line indicates a moving average over 100 rounds, blue line a moving average over 1000 rounds; vertical dashed lines indicate introduction of a new signal; to keep the size of the pdf graphic in check, only values for every 10th iteration are shown. Notice how communication quality lingers somewhere around 0.1 until the seventh signal is introduced, where it suddenly increases dramatically.

as pronounced as shown in figure 3. In fact, the sigmoid curves of canonical games are at times so flat that one might mistake them for linear straights.

What may be a cause of this behaviour? When new signals are introduced into an ongoing game, then upon introduction every sender receives one ball corresponding to the new signal into each of their urns and every receiver receives a new urn with ordinary initial configuration corresponding to the new signal. Somehow this mechanism seems to have disrupted emerging communication for a good thirty thousand iterations, but then communication emerges suddenly and rapidly even though new signals keep being introduced to the game up until the 42000th iteration—even after the game has passed the convergence threshold. To seriously investigate this phenomenon, it is most likely necessary to keep track of the contents of every urn and develop several useful metrics to capture the effects of signal introduction. While possible in principle, collecting such data and implementing such metrics is beyond the scope of this essay.

Micro scale

The linear relationship of figure 2 has proven to be of little interest. In a second set of simulations the values chosen for `signals-interval` were much smaller. The following values are those with which the model was run:

```
population-size = 1
num-remove-balls = 1
num-add-balls = 2
num-world-states = 10
```

`signals-interval` was again increased step-wise, but this time on much smaller scales. First, from 1 to 100 (with one hundred games being performed for every value) and then secondly from 1 to 10 (with one thousand games being performed for every value). The results are plotted on the left and right hand side of figure 4 respectively.

On the left plot, a linear relation between signal introduction and convergence speed is once again visible. Statistics affirm this visual hunch with $R^2 = 0.76$. The right-hand plot shows a somewhat more interesting picture. The increase in convergence time experiences a drop for `signals-interval = 7` that goes even below the convergence speed for `signals-interval = 0` (i.e. all signals are available from the beginning of the game on). Unfortunately, testing the distribution of convergence speeds with `signals-interval = 7` for significantly low values against the distribution with `signals-interval = 0` results in $p = 1.0$ with both a Student-t test and a Wilcoxon signed-rank test (the latter does not assume the data to be normally distributed; some high outliers would in fact have to be excluded for the data to pass tests of normality, e.g. Shapiro-Wilk).

3.3 Reproducing results by Barrett (2006)

Reproducing existing (and potentially widely established) results is always a desirous endeavour in empirical science. The present essay is by far not the first simulation approach to LSGs, so this section aims to reproduce (or fail to reproduce) results already existing in the literature.

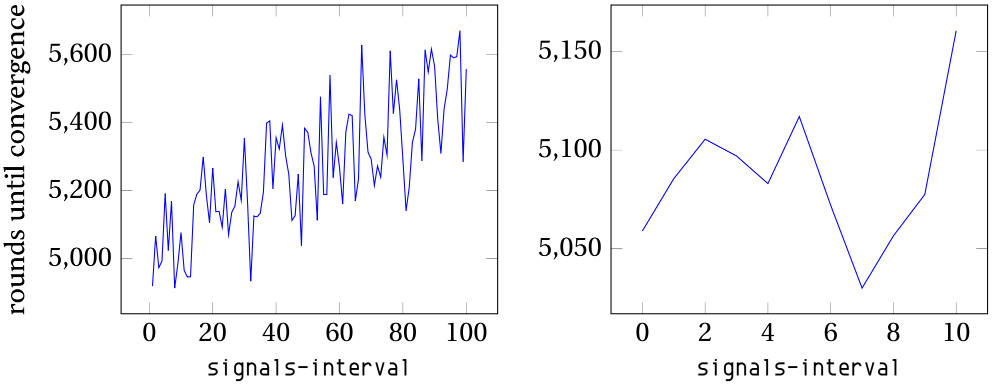

Figure 4: Effects of later introduction of signals on a micro scale; x-axis indicates value of `signals-interval`, y-axis indicates median iterations until convergence to perfect communication. *Left:* `signals-interval` incremented from 0 to 100 with 100 games played per value (a value of 0 indicating all signals being available from the beginning on), $R^2 = 0.76$. *Right:* `signals-interval` incremented from 1 to 10 with 1000 games played per value, $R^2 = 0.04$.

In Barrett (2006), the author introduces the notion of an *n*-state/*n*-term LSG, in which there are *n* possible states of the world and *n* possible actions the receiver may choose to perform. Via simulation, Barrett shows that 2-state/2-term LSG always converge to perfect communication, but that the the same is not true for 3-state/3-term LSGs, 4-state/4-term LSGs, and 8-state/8-term LSGs (or in all probability any values for $n > 2$). In his experiments, Barrett ran one thousand games for every value of *n* and then calculated the ratio of games that had converged to perfect communication. Table 1 shows Barrett's results compared to mine. While the results are identical for 2-state/2-term LSG, the same is not true for other values of *n*. In an effort to rule out chance as a possible cause for these vast discrepancies, I have rerun the experiment for 2-state/2-term LSGs and performed 20000 games. But even this second, much longer, rerun resulted in a ratio of 0.9735 of games converging to perfect communication.

Another possible cause besides chance that comes to mind are subtleties in the definition of convergence to perfect communication. Barrett (2006, sec. 2, unpaginated preprint) writes, that "[o]ut of 10^3 runs, there were three where the success rate was less than 0.8 after 10^6 plays" and further that "[t]here are 10^6 plays/run here, and a run is taken to fail to converge if the final signal success rate of the run is less than 0.8." This leaves some room open for interpretation however.

	success rate		
model	Barrett	mine	difference
2-state/2-term	1.000	0.999	- 0.001
3-state/3-term	0.096	0.974	+ 0.878
4-state/4-term	0.219	0.933	+ 0.714
8-state/8-term	0.594	1.000	+ 0.406

Table 1: Percentage of games leading to convergence in an *n*-state/*n*-term LSG with 1000 games played per run; values for Barrett's (2006) and my own simulations respectively.

If Barrett's words are to be taken literally, then the measure of convergence takes into account only the one millionth round and ignores everything that happened beforehand. But the variance of communication quality in between rounds is vast. In figure 3, the light grey line indicates a moving average over the past one hundred rounds played. Yet even this moving average shows drastic fluctuations, especially towards convergence where fluctuations reach an amplitude of about 0.1. A closer look at games failing to converge shows that this fluctuation does not stop, not even decrease, as the rounds progress. And again, the fluctuations visible in figure 3 are of a moving average over the past one hundred rounds. The fluctuations in communication quality in each individual round are undoubtedly much higher. It therefore seems unreasonable that Barrett defines convergence in such a way that only a single round is taken into account. Chance would drastically influence the ratio of games that converge, and this is surely not a desired characteristic for a metric. It could, however, potentially explain the much lower values of Barrett's experiments. Many games that would be ruled out by chance when considering only a single round would pass the convergence test when looking at a moving average over the past one thousand rounds, as was the case in the present essay (for the precise definition employed here, see paragraph on observation in subsection refssec:moddes).

There is one more subtlety that needs to be mentioned here. For the experiments in table 1 only, I have considered games as non-convergent if they had not converged by the 100000$^{\text{th}}$ iteration; the simple reason being the amount of time the simulations took. But this cannot be the cause of the discrepancy for two reasons: First off, the amount of games "incorrectly" classified as non-convergent is in all probability very low. The median convergence time for all *s*-state/*s*-term LSG run is below two thousand, with no game having converged after more than thirty thousand iterations. Secondly, *if* a game has been incorrectly classified as non-convergent, then this would of course distort the data but it would result in values *lower* than the ones I was supposed to obtain. In other words, if at all influenced by substituting one hundred thousand for one million, then this inaccuracy has decreased the gap in table 1, not increased it. Therefore, it cannot possibly be an explanation for it.

Nevertheless, despite the different ratios the key findings of Barrett (2006) have been replicated. It is the case that 2-state/2-term LSGs converge with a ratio of 0.999 and it is the case that *n*-state/*n*-term LSGs with $n > 2$ do converge in some cases but do not converge in others. Another conclusion of Barrett (2006, sec. 2, unpaginated preprint) has also been reproduced: "While failures to approach signaling systems were observed, each system always learned to do better than chance in signaling, and hence might be said to have developed a more or less effective language." While the exact values of communication quality for non-convergent games has not been recorded for the present simulations, communication quality did always go above chance (with chance in an *n*-state/*n*-term LSG being equal to a ratio of $\frac{1}{n}$).

Other results of Barrett (2006) have also been reliably reproduced. Table 2 shows the contents of both the sender's and the receiver's urn in an 3-state/3-term LSG after one million iterations. In both Barrett's and my case a near-perfect signalling system has emerged. Note that the important characteristic here is the fact that in every urn the vast majority of balls correspond to the same signal or action. Which of the urns has converged to which signal or action is in principle irrelevant as long as the mapping from states of the world to signals to actions to states of the world is correct. A signalling system does not intrinsically distinguish between signal 0 and signal 1 or action A and action B.

There is again, however, a subtle difference in the results of table 2. Notice how in Barrett's urns, the number of balls not corresponding to the most frequent signal or action is sometimes 1 but sometimes slightly higher than 1. In contrast, the number of balls not corresponding to the most frequent signal or action in the urns of the present simulation is always 1. Although the precise cause of this difference has eluded me, it may be hinting at why Barrett's simulations converged so much less than mine. 1 is in fact the lowest value attainable in the data, because the learning

SENDER			RECEIVER		
# of balls	Barrett	mine	# of balls	Barrett	mine
state A urn			*signal 0 urn*		
signal 0	3	1	act A	3	1
signal 1	333688	332968	act B	1	333599
signal 2	2	1	act C	333141	1
state B urn			*signal 1 urn*		
signal 0	1	333601	act A	333688	332971
signal 1	1	1	act B	1	1
signal 2	332514	1	act C	4	1
state C urn			*signal 2 urn*		
signal 0	333141	1	act A	2	1
signal 1	4	1	act B	332514	1
signal 2	20	333299	act C	20	333298

Table 2: Contents of the sender's and the receiver's urns in a 3-state/3-term LSG after one million iterations in Barrett's (2006) and my own simulations respectively.

function is defined in such a way that at least one ball with a given signal or action always remains in the urn (see paragraph on learning in subsection 2.4), and can therefore practically be read as zero.

4 Conclusion

The few simulations carried out in the previous section are of course only the tip of the iceberg of what the model is capable of. Throughout the simulations I have kept most of the parameters constant; not because I do not deem them useful, but simply because the time it takes to properly explore the model's entire parameter space is more adequately measured in days than hours. Consequently, the open space for further exploration of the model is vast. A desirable discovery would be parameter settings that speed up the convergence towards perfect communication. In the simulations performed, all parameter changes that deviated from the "default" have slowed down the convergence speed.

The unsuccessful attempts to reproduce some of the results of Barrett (2006) are in all probability caused by a subtle implementational detail, that differs between Barrett's and my implementation. But without a detailed description of Barrett's (2006) model, which his papers lack, finding said detail is almost an impossible feat.

References

BARRETT, Jeffrey A. (2006): “Numerical simulations of the lewis signaling game: Learning strategies, pooling equilibria, and the evolution of grammar.” *UC Irvine Institute for Mathematical Behavioral Sciences* preprint.

BARRETT, Jeffrey A. (2007): “Dynamic partitioning and the conventionality of kinds.” *Philosophy of Science* **74** (527–546).

GRIMM, Volker; BERGER, Uta; BASTIANSEN, Finn; ELIASSEN, Sigrunn; GINOT, Vincent; GISKE, Jarl; GOSS-CUSTARD, John; GRAND, Tamara; HEINZ, Simone K.; HUSE, Geir; HUTH, Andreas; JEPSEN, Jane U.; JØRGENSEN, Christian; MOOIJ, Wolf M.; MÜLLER, Birgit; PE’ER, Guy; PIOU, Cyril; RAILSBACK, Steven F.; ROBBINS, Andrew M.; ROBBINS, Martha M.; ROSSMANITH, Eva; RÜGER, Nadja; STRAND, Espen; SOUISSI, Sami; STILLMAN, Richard A.; VABØ, Rune; VISSER, Ute; and DEANGELIS, Donald L. (2006): “A standard protocol for describing individual-based and agent-based models.” *Ecological Modelling* **198**: 115–126.

GRIMM, Volker; BERGER, Uta; DEANGELIS, Donald L.; POLHILL, J. Gary; GISKEE, Jarl; and RAILSBACK, Steven F. (2010): “The ODD protocol: A review and first update.” *Ecological Modelling* **221**: 2760–2768.

GRIMM, Volker and RAILSBACK, Steven F. (2005): *Individual-Based Modeling and Ecology*. Princeton, NJ: Princeton University Press.

HEATH, B. L. and HILL, R. R. (2014): “Some insights into the emergence of agent-based modeling.” In: TAYLOR, Simon Je, ed., “Agent-based modeling and simulation,” 32–44. New York, NY: Palgrave MacMillan.

LEWIS, David (1969): *Convention*. Cambridge, MA: Harvard University Press.

MÜLLER, Birgit; BOHN, Friedrich; DREISSLER, Gunnar; GROENEVELD, Jürgen; KLASSERT, Christian; MARTIN, Romina; SCHLÜTER, Maja; SCHULZE, Jule; WEISE, Hanna; and SCHWARZ, Nina (2013): “Describing human decisions in agent-based models – ODD + D, an extension of the ODD protocol.” *Environmental Modelling & Software* **48**: 37–48.

RAILSBACK, Steven F. and GRIMM, Volker (2011): *Agent-based and individual-based modeling: A practical introduction*. Princeton, NJ: Princeton University Press.

WILENSKY, Uri (1997): *NetLogo ant lines model*. Evanston, IL: Center for connected learning and computer-based modeling, Northwestern University.
URL: `http://ccl.northwestern.edu/netlogo/models/AntLines`

WILENSKY, Uri (1999): *NetLogo*. Evanston, IL: Center for connected learning and computer-based modeling, Northwestern University.
URL: `http://ccl.northwestern.edu/netlogo/`