1) Automata Description - Provide a diagram or a clear written description of your automaton. Include the cooperation probability for each state and the transition rules for all outcomes (CC, CD, DC, DD). Explain the specific purpose of each state (e.g., "State 0 is the initial cooperative state," "State 1 is for retaliation").

State 0 - This is the initial cooperative state meant to always cooperate so long as the opponent is cooperating.

Cooperation probability: 1.00
CC: State 0
CD: State 1
DC: State 0
DD: State 0

State 1 - This state seeks to mostly cooperate, but is also slightly suspicious of the opponent as a defect was previously detected despite our cooperation. We want to operate at CC as much as possible, so seeing a CC will return us to State 0.

Cooperation probability: 0.75
CC: State 0
CD: State 2
DC: State 1
DD: State 1

State 2 - This is a more suspicious state which also has the option to return back to State 0 if a CC is found, but we are less likely to cooperate as a result of a burgeoning pattern of defects from the opponent. In this state, we will always become less trustful (move to State 3) if the opponent defects, regardless of whether or not we also defect.

Cooperation probability: 0.33
CC: State 0
CD: State 3
DC: State 2
DD: State 3

State 3 - This is a slightly more forgiving iteration of the “grim trigger” state discussed in class. It is unlikely for us to cooperate, but if the opponent does so, then we will begrudgingly move back into State 2 to remain suspicious, but still give a possible route to return to full cooperation in the event that a string of noise causes several defects in quick succession.

Cooperation probability: 0.01
CC: State 2
CD: State 3
DC: State 2
DD: State 3

2) Strategy Philosophy - Describe the core philosophy of your agent. Is it aggressive, forgiving, or random? How does it attempt to maximize score in a noisy environment?

Our agent seeks to operate at CC as often as possible in order to maximize our own score without building a negative reputation. In order to do so, we cooperate until the opponent defects, then are fairly quick to forgive if an outlier defect (or one caused by random noise) occurs. This allows us to bounce back from the small probability that noise interrupts a symbiotic relationship with another cooperative agent while also protecting against hostile opponents that are unforgiving or show numerous defections for other reasons.

3) Noise Resilience - Explain specifically how your agent handles the 5% noise. If your agent accidentally defects (or perceives a defection that didn't happen), how does it recover cooperation? Or does it not care?

As stated above, our agent seeks to operate at CC as often as possible. In order to do so, we have to be forgiving ourselves while also proving to the opponent that we are trustworthy. This is why our 4 state machine acts as a sort of "scale" of cooperation and trust. As more defects are detected from the opponent, we get increasingly more suspicious in order to protect our own score, but we also seek to bring back full cooperation if the opponent demonstrates a willingness to do so as well. As such, noise causing an opponent to switch to defecting forever will be protected against by us moving into State 3 while noise will not force our machine into always defecting immediately but rather a slightly less cooperative state than the always cooperative State 0.

4) Opponent Analysis - How does your agent fare against a pure "Always Defect" strategy? Does it get exploited, or does it protect itself? How does it fare against "Tit-for-Tat" in a noisy environment?

Our agent protects itself against Always Defect. It takes a few turns, but eventually our machine will enter State 3 and begin to defect 99% of the time in order to minimize the damage caused by Always Defect.

Against Tit-for-Tat, a cooperative relationship emerges at first, but noise will eventually interrupt it. Since our machine is forgiving and Tit-for-Tat only considers the most recent move our agent played, it is easy for us to get back on track to a fully cooperative state again despite noise causing minor hiccups along the way.

5) Self-Play - What happens when your agent plays against itself? Do they cooperate effectively, or do they spiral into mutual defection due to noise?

Our agent generally performs well. With the ability to transition back into a fully trusting state, random noise typically doesn't spiral the agents into mutual destruction. In most cases, after a noisy defect, each agent may respond with a defect in response, but any sign of cooperation will restore it to a more trusting state.

In larger test cases, the agents do sometimes reach mutual defection, but our automata does create the possibility of escaping this mutual defection, through a state that is not unlike optimistic unblocking in P2P systems. When both are trying to reach a more "trusting" state, and give opportunity to cooperate, it works well.

6) Tournament Prediction - If the class consists mostly of "Tit-for-Tat" variants, how will your agent perform? What if the class is mostly "Always Defect"?

In the class with mostly "Always Defect" we will not perform as well, but limit our losses quickly, reaching the mutually defect state almost immediately. Our optimistic cooperation 1% of the time in state 4 does create additional losses against uncooperative agents, which widens the gap against some of these agents. Additionally, against always defect agents, both scores are limited, and both scores will be relatively low in the set of possible scores. In a class of all "always defect", we will certainly lose, but we hope to bridge this gap in score playing against any cooperative agents.

If the class consists of mostly tit for tat variants, the agent will perform moderately well, with tests coming in at an average around 2 points per round. Noisy defects might throw us into a cascading spiral, but oftentimes our several trust thresholds will restore mutual cooperation. We perform almost identical score outputs against tit-for-tat. Our scores improved dramatically against our own variations on tit-for-tat, and I trust against the class it will perform in a similar fashion.

7) Exploitation - Design a theoretical 'Anti-MyAgent' that would perfectly exploit your strategy. What would it do?

I theoretical Anti-MyAgent would take advantage of our strategies' forgiving nature and do infrequent defects and build back up cooperation to State 0 before defecting again. A sample strategy could use many more states, and cooperate until some time t, where they defect, and then resume cooperating. Any repeated defects would transition us to State 3, but this sort of strategy would going back and forth from State 0 and State 1, maximizing points by lone defects and otherwise mutual cooperation.

8) Parameter Tuning - If you used probabilistic transitions (e.g., 0.8), how did you tune these numbers? What happens if you change them?

Tuning these numbers was a tricky process. Our first thought was to maximize utility with probabilistic transitions given the sample agents, but it is likely the class will offer some variations on these agents that our numbers aren't tuned to. Instead of an optimization approach, we largely considered the human psychology of the class and what we expected our classmates to do. A lot of the literature on this problem points to cooperation being optimal in the long run, so we tuned our parameters to be more inclined to cooperation in all states.

Particularly, in State 3, we settled on some probability 0.01 of cooperation, which against strong-defect models, loses us a lot of points. We acknowledged this, but decided to look past this because it offered greater point totals in cooperative environments where noise derails cooperation. This strategy was largely inspired by the optimistic unblocking found in the BitTorrent model that encourages cooperation in the P2P model. We thought that this sort of strategy offered maximum utility given the tendency towards cooperation in the long run, and the lost utility was minimal.