

Whence tit-for-tat?

PHILIP H. CROWLEY* and R. CRAIG SARGENT

Center for Ecology, Evolution and Behavior and T.H. Morgan School of Biological Sciences, University of Kentucky, Lexington, KY 40506-0225, USA

Summary

In theoretical and empirical studies of the evolution of cooperation, the tit-for-tat strategy (i.e. cooperate unless your partner did not cooperate in the previous interaction) is widely considered to be of central importance. Nevertheless, surprisingly little is known about the conditions in which tit-for-tat appears and disappears across generations in a population of interacting individuals. Here, we apply a newly developed classifier-system model (EvA) in addressing this issue when the key features of interactions are caricatured using the iterated prisoner's dilemma game. Our simple representation of behavioural strategies as algorithms composed of two interacting rules allowed us to determine conditions in which tit-for-tat can replace non-cooperative strategies and vice versa. Using direct game-theoretic analysis and simulations with the EvA model, we determined that no strategy is evolutionarily stable, but larger population sizes and longer sequences of interactions between individuals can yield transient dominance by tit-for-tat. Genetic drift among behaviourally equivalent strategies is the key mechanism underlying this dominance. Our analysis suggests that tit-for-tat could be important in nature for cognitively simple organisms of limited memory capacity, in strongly kin-selected or group-selected populations, when interaction sequences between individuals are relatively short, in moderate-sized populations of widely interacting individuals and when defectors appear in the population with moderate frequency.

Keywords: classifier system; evolution of cooperation; game theory; genetic algorithm; iterated prisoner's dilemma; reciprocal altruism

Introduction

Who shall dwell in thy holy hill?
He that . . . worketh righteousness,
and speaketh the truth . . .
In whose eyes a vile person is condemned;
He that sweareth to his own hurt, and changeth not . . .
nor taketh reward against the innocent.

Psalms 15:1–5

There is much current interest in understanding how cooperation can evolve in populations of organisms under the influence of natural selection. Three main mechanisms involving direct interactions among individuals have been proposed: group selection (Wilson, 1980; Mesterton-Gibbons and Dugatkin, 1992), kin selection (Hamilton, 1964; West Eberhard, 1975) and reciprocal altruism (Axelrod and Hamilton, 1981; Axelrod, 1984; Trivers 1985). [By-product mutualism, an indirect interaction among individuals mediated by their environment, has also been proposed as a mechanism for evolving cooperation (Brown, 1987; Mesterton-Gibbons and Dugatkin, 1992), but we will not consider it further here.] Group selection may be important within certain metapopulations, especially those with sufficiently isolated subpopulations more likely to persist

* To whom correspondence should be addressed.

and export colonists when they achieve high absolute fitnesses; however, such a metapopulation structure may be uncommon (e.g. see Harrison, 1991) and individual selection may often overwhelm the effects of group selection in nature (Williams, 1966). In many, perhaps most, cooperative populations, kin selection – in which relatives gain fitness both from their own success and that of their kin – may be important, at least initially (Axelrod and Hamilton, 1981). However, there are many other well-documented examples that seem to fit neither group nor kin selection; these may hinge primarily on behavioural reciprocation of actions beneficial to the recipient over a sequence of interactions between individuals (Axelrod and Hamilton, 1981; Axelrod, 1984; Axelrod and Dion, 1988; Crowley *et al.*, 1995). The intriguing possibility that cooperation may readily evolve in the absence of the other mechanisms motivates us to focus here on reciprocal altruism.

A particularly useful tool for investigating the evolution of cooperation by reciprocal altruism is the iterated prisoner's dilemma (IPD) game (Fig. 1; Axelrod and Hamilton, 1981). In the present context, we focus our analysis of reciprocal altruism on the frequency of mutual cooperation between the two IPD players. This can be interpreted as simultaneous reciprocal altruism, in contrast to Trivers' (1971) initial concept, in which the two players alternate cooperation and defection out of phase with each other.

The two competitions conducted by Robert Axelrod (University of Michigan) to identify the most successful competitive yet cooperative strategies in playing the IPD produced the same winner both times: tit-for-tat (TFT) (see Axelrod, 1984; Axelrod and Dion, 1988). TFT cooperates in the first interaction of an IPD sequence and then in subsequent interactions simply does what its partner did last. The personality of TFT has been characterized (Axelrod and Dion, 1988) as nice

		Focal's Response	
		C = Cooperate	D = Defect
Other's Response	C	3 R = Reward	5 T = Temptation
	D	0 S = Sucker	1 P = Punishment

Figure 1. Fitness pay-offs to the focal individual for each of the four possible response combinations in the prisoner's dilemma game. (Pay-offs to the other individual are the same as for the focal, except that when the focal receives T, the other receives S and vice versa.) The game is defined by inequalities among the pay-offs, namely $T > R > P > S$ and $2R > S+T$. Regardless of the other's response, the focal's best (fitness maximizing) move is to defect. A sequence of such games is known as the iterated prisoner's dilemma (IPD) and in general there is no best strategy in the IPD (Boyd and Lorberbaum, 1987; Mesterton-Gibbons, 1992). Two selfish individuals can establish a rewarding relationship only by each taking the risk of being suckered, which neatly caricatures the problem faced by many organisms that somehow manage to cooperate in nature. Memory of previous responses opens the possibility of reciprocity and this can sometimes steer an interaction towards cooperation.

(never defects first), provokable (defects after defections by the partner), forgiving (cooperates after the partner cooperates, regardless of what went before) and clear (straightforward and consistent) [cf. the above passage from the Book of Psalms].

TFT (and some close relatives) has become the prototypical cooperative strategy in both empirical studies (Milinski, 1987, 1990; Dugatkin, 1991; Dugatkin and Alfieri, 1991; and others reviewed in Axelrod and Dion (1988) and Crowley *et al.* (1995)) and theory (e.g. Axelrod, 1987; Boyd and Lorberbaum, 1987; Mesterton-Gibbons and Dugatkin, 1992; Nowak and Sigmund, 1992) of reciprocal altruism. Despite its laurels in the tournaments and other documented evolutionary successes of its close relatives (Axelrod, 1987; Nowak and Sigmund, 1992), however, TFT is unable to do better than its partner over a sequence of interactions or to avoid the consequences of its obligation to retaliate, which may sometimes result in its replacement by more sophisticated strategies in evolutionary simulations based on the IPD (Lindgren, 1991; Nowak and Sigmund, 1992, 1993; Crowley, 1995; Crowley *et al.*, 1995). Nevertheless, TFT may at least have an important role as an intermediate step (Lindgren, 1991) or a 'catalyst' (Nowak and Sigmund, 1992) in the evolution of cooperation. Thus, understanding how TFT may evolve and the conditions under which it can persist for extended periods may prove quite valuable in approaching the more general question of how cooperation is initiated and maintained from an initially asocial state (Elster, 1979).

Axelrod and Hamilton (1981) argued that TFT may arise either through kinship or through clustering, in which TFT players are, for whatever reason, more likely to interact with each other than with other strategies. However, neither of these mechanisms is required for TFT to evolve under conditions in which the IPD is played by simple, evolving algorithms using memory limited to the previous play of the game (Crowley, 1995). This previous paper described a classifier-system model of the evolution of cooperation (EvA) capable of tracking these algorithms (= strategies) across simulated evolutionary time. In the present study, we use this model to trace the evolutionary dynamics of TFT and develop an understanding of some of the forces that determine how cooperative strategies may arise and disappear.

In the remainder of this paper, we describe the EvA model and then use it to express a simple class of strategies that includes TFT in algorithmic form. An analysis of all possible algorithms in this class and their mutational transitions then suggests the importance to the evolution of cooperation of the population size and of the number of interactions per generation between each two individuals. We therefore run the model on a computer to determine the effects of population size and interactions per pair per generation on fitness and on the percent of all interactions resulting in mutual cooperation, and we examine a representative sequence of generations over which TFT appears and disappears. From these results, we summarize the evolutionary dynamics of TFT and some other simple strategies, make some testable predictions and identify some important directions for future research.

The EvA model

Classifier systems (Holland, 1975, 1992; Goldberg, 1989), closely related to genetic algorithms (Axelrod, 1987; Sumida *et al.*, 1990; Davis, 1991; Crowley *et al.*, 1995), evolve algorithms that generate an action in response to a condition; in the present context, the action would be cooperation (C) or defection (D) in a round of the IPD and the condition would be the responses by the focal individual (i.e. the algorithm in question) and its opponent in the immediately preceding round. These evolving algorithms, which can be interpreted as strategies, are sets of interacting rules, each occupying a locus on the genome of the individual (= algorithm). A summary of the grammar of EvA, as it pertains to the present analysis, is presented in Fig. 2. The

basic premisses of EvA are that behaviours can be considered to consist of logical, interacting components, that behaviours develop from related behaviours and that the most successful behaviours tend to become more common. Success or fitness is measured by the accumulation of pay-offs from playing the IPD against all other individuals in the population.

Transitions between algorithms differing by a single-point mutation

To investigate the dynamics of TFT, we simplify the problem by considering only the 35 possible two-rule algorithms (Table 1). This assemblage includes TFT and an array of other relatively simple strategies (see Fig. 2) that underpin the more sophisticated behaviour of much more

Table 1. Single-mutation transitions among the 35 possible two-rule algorithms and the potential for successful invasion^a

Second rule	First rule		Algorithm number ^b	Single-mutation transitions	Algorithm number ^b	Single-mutation transitions
	:/C	:/D				
:/C	1	2 3 5 7 9	-	-	-	-
:/D	2	1 4 6 8 10 19 20 22 24 26	[19]	2 21 23 25 27		
C:/C	3	1 4 5 11 13 20	[20]	2 3 21 22 28 30		
C:/D	4	2 3 6 12 14 21	[21]	4 19 20 23 29 31		
D:/C	5	1 3 6 15 17 22	22	2 5 20 23 32 34		
D:/D	6	2 4 5 16 18 23	[23]	6 19 21 22 33 35		
/C:C	7	1 8 9 11 15 24	[24]	2 7 25 26 28 32		
/C:D	8	2 7 10 12 16 25	[25]	8 19 24 27 29 33		
/D:C	9	1 7 10 13 17 26	26	2 9 24 27 30 34		
/D:D	[10]	2 8 9 14 18 27	[27]	10 19 25 26 31 35		
C/C:C	11	3 7 12 13 15 28	[28]	11 20 24 29 30 32		
C/C:D	12	4 8 11 14 16 29	[29]	12 21 25 28 31 33		
C/D:C	13	3 9 11 14 17 30	[30]	13 20 26 28 31 34		
C/D:D	14	4 10 12 13 18 31	[31]	14 21 27 29 30 35		
D/C:C	[15]	5 7 11 16 17 32	[32]	15 22 24 28 33 34		
D/C:D	16	6 8 12 15 18 33	[33]	16 23 25 29 32 35		
D/D:C	17	5 9 13 15 18 34	34	17 22 26 30 32 35		
D/D:D	18	6 10 14 16 17 35	[35]	18 23 27 31 33 34		

complex strategies (Crowley, 1995). Moreover, the basic patterns and mechanisms that emerge here generally characterize those associated with algorithms based on larger rule sets and rules invoking more memory (Crowley, 1995).

The mutations considered here are single-point modifications: changes between C and D and insertions and deletions of a randomly chosen C or D within rules. These mutations are generated with sufficient frequency to prevent any of the algorithms or groups of algorithms from being evolutionarily stable (see Boyd and Lorberbaum, 1987), but not so frequently that dominance of one or more algorithms cannot be maintained over tens or hundreds of generations by selection. By considering the fate of a population composed entirely of each individual algorithm invaded by one of its single-point mutants, we can begin to understand the quasi-stability punctuated by evolutionary transitions between algorithms (Appendix 1 and Table 1). The analysis in Appendix 1 shows that TFT is the only cooperative algorithm that cannot be invaded by any of its single-point mutants. In contrast, none of the 12 always defect (ALL D) algorithms can be invaded by a single-point mutant. Under assumptions of no more than one point mutation per population per generation and no genetic drift, then both TFT and ALL D would be ESSs. So how do TFT and ALL D ever successfully invade each other?

Dynamics of invasion and replacement

with the exception of extremely high mutation rates, which are not considered here, successful invasion and elimination of TFT by ALL D and vice versa requires a combination of mutation and drift (Appendix 2). A TFT population becomes invisable by an ALL D mutant once approximately half of the TFT algorithms have drifted to always cooperate (ALL C) (see Table 2); with fewer ALL C present than this critical proportion, a mutant ALL D will actually 'inoculate' the population against such invasion by selecting for TFT over the exploitable ALL C. When most of the cooperators are TFT, the advantage that ALL D gets from exploiting the few ALL C is outweighed by reciprocation among TFT; this results in the elimination of both ALL D and the exploited ALL C, re-establishing TFT fixation. Thus, rapid replacement of TFT by ALL D would require a reasonably high mutation rate, so that the process is not simply retarded by the rarity of mutations, but drift must then be especially rapid to permit accumulation of sufficient ALL C between appearances of mutant ALL D.

^a Single-mutation transitions are the algorithms into which the given algorithm can mutate by point mutation within a rule. Symbols above these mutant algorithms indicate their expected fate in subsequent generations. Consider algorithms A and B. \hat{A} means that a mutant algorithm A will invade and replace algorithm B and that a mutant algorithm B is unable to invade A and will disappear. \hat{B} implies that a mutant A cannot invade B and will disappear, whereas a mutant B will invade and replace A. \hat{A} indicates that a mutant A cannot invade B, but neither can a mutant B invade A. \hat{A} implies that a mutant A can invade B and will dominate a stable mixture of A and B; moreover, a mutant B can invade A but will be dominated by A in the stable mixture. Finally, \hat{A} means that a mutant A can invade B and will be dominated by B in a stable mixture of A and B and a mutant B can invade A and will dominate A in the stable mixture. The algorithm identifier A with none of these additional symbols on top means that mutant A has neither advantage nor disadvantage relative to algorithm B and the same for mutant B relative to A: changes in their frequencies will result solely from genetic drift.

^b The boxes around some of the algorithm numbers imply non-invasibility of that algorithm by any of its single-mutation transition algorithms. Note that 12 of the 18 /:D algorithms are non-invasible, whereas only one /:C algorithm is non-invasible, namely TFT. Three others (indicated by square parentheses) can be invaded but not completely eliminated by algorithms derived from single-point mutations. Also note that all of the functionally identical ALL D algorithms (i.e. both rules result in defection: 19, 21, 23, 25, 27, 29, 31, 33 and 35) are non-invasible; in contrast, all ALL C algorithms are invisable and most (i.e. 1, 3, 5, 7, 9, 11, 13 and 17) are eliminated by the invader, except for algorithm 15, which persists as a minority component of a mixture with its only successful invader, algorithm 32 (see Appendix 1).

Invasion of ALL D by TFT requires specific precursors [i.e. suspicious TFT (STFT) and latent retaliator; see Table 2]. Latent retaliator is an ALL D algorithm that can accumulate by mutation and drift, and an individual of this type must mutate to yield the TFT invader. STFT can also invade an ALL D population by mutation and drift; TFT will proceed to eliminate ALL D in the presence of even a single STFT for most situations of interest here. Exceptions arise when the interaction sequences are quite short and populations are relatively large; however, for a sequence length of 11, the frequency of STFT required for successful invasion of ALL D by TFT is less than 1/15.

We summarize our understanding of transitions between TFT and ALL D, based on the analyses of Appendices 1 and 2, in Fig. 3. This interpretation led us to expect that a large population size, which reduces the rate of genetic drift (Wright, 1921) and long interaction sequences, which facilitate reciprocation (Axelrod and Hamilton, 1981), would jointly shift the balance of behaviour

RULE SYNTAX *X/Y:Z* *Translation:* If the focal individual did X and the other did Y in the last round, then the focal does Z in this round.

<u>Example Rule</u>	<u>Translation</u>
/:C	No matter what either partner did last, the focal cooperates
/C:D	If the other cooperated in the last round, the focal defects
D/C:C	If the focal defected and the other cooperated last, the focal cooperates

RULE SEMANTICS: How rules interact to determine the response in a two-rule algorithm.

- (1) When a more specific rule fits, it overrides a more general rule.
- (2) When equally specific rules both fit, they are equally likely to be invoked (i.e. EQUIVOCATOR, the combination of /:C and /:D, yields C or D, each with probability 0.5).
- (3) There is always at least one maximally general rule per algorithm (i.e. /:C or /:D).

EXAMPLE INTERACTION SEQUENCE FOR ALGORITHMS $A = \begin{cases} /:C \\ C/D:D \end{cases}$ & $B = \begin{cases} /:D \\ /D:C \end{cases}$
 (Payoffs indicated in square brackets come from Figure 1)

ROUND:	1	2	3	4	5	6
ALGORITHM A:	C [0]	D [1]	C [3]	C [0]	D [1]	C [3]
ALGORITHM B:	D [5]	D [1]	C [3]	D [5]	D [1]	C [3]

Note that the sequence of responses in the first 3 rounds simply repeats indefinitely, and B achieves the higher total payoff (fitness).

SOME IMPORTANT TWO-RULE ALGORITHMS

/:C	/:D	/:D	/:C	/:D	/:C	* = C, D, or neither
/:C	*/*:D	/D:D	/D:D	/C:C	/:D	
ALL C	ALL D	LATENT RETALIATOR	TFT	STFT	EQUIVOCATOR	

Figure 2. Brief overview of the grammar of EvA for two-rule algorithms. TFT, tit-for-tat; STFT, suspicious tit-for-tat.

Table 2. Minimal numbers of STFT (s^*) and ALL C (c^*) needed for TFT to invade ALL D and for ALL D to invade TFT, respectively, for the combinations of population size and sequence length used in the EvA runs reported in the text^a

	Population size ($n + 1$)								
	10			20			40		
Sequence length (m)	11	31	101	11	31	101	11	31	101
s^* for TFT to invade ALL D	1	1	1	2	1	1	3	1	1
c^* for ALL D to invade TFT	4	4	4	8	9	9	17	19	19

^a See Appendix 2.

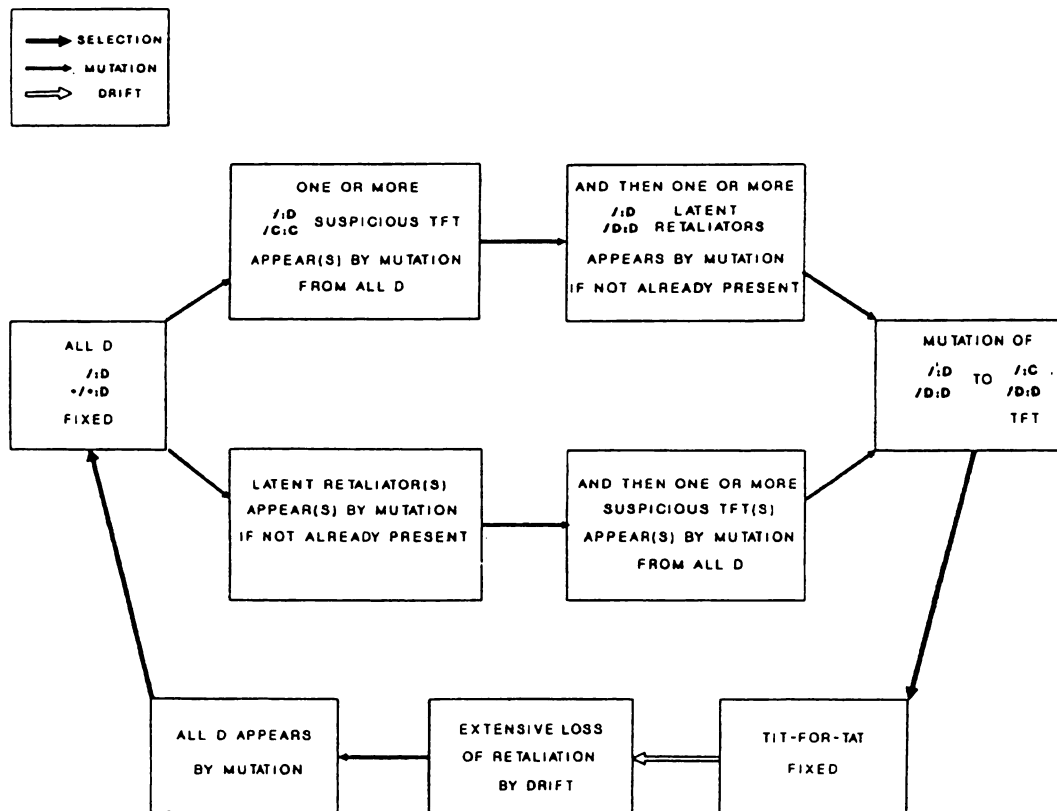


Figure 3. How selection, mutation and drift generate the evolutionary dynamics of TFT. Before TFT can invade ALL D, defectors that can cooperate in response to cooperation (here, only STFT) and other individuals bearing a latent retaliation rule must appear, though these events can occur in either order. Eventually, a latent retaliator mutates into TFT, which will quickly spread throughout the population if the interaction sequences are long enough. ALL C can then gradually begin to accumulate in the population through genetic drift. Ultimately, enough ALL C individuals will be present that a mutant ALL D will be able to invade and spread. See Appendix 2.

Table 3. The appearance and disappearance of tit-for-tat in a run of the EvA model¹⁸

Generation number	Mean fitness	% mutual cooperation	Algorithms (i.e. rule pairs)
136	1.000	0.0	$\left. \begin{array}{l} /:D \\ /D:D \end{array} \right\} (15)$ $\left. \begin{array}{l} /:D \\ C/D:D \end{array} \right\} (3)$ $\left. \begin{array}{l} /:D \\ /C:D \end{array} \right\} (1)$ $\left. \begin{array}{l} /:D \\ /:D \end{array} \right\} (1)$
140	1.000	0.0	$\left. \begin{array}{l} /:D \\ C/D:D \end{array} \right\} (8)$ $\left. \begin{array}{l} /:D \\ /C:D \end{array} \right\} (6)$ $\left. \begin{array}{l} /:D \\ /D:D \end{array} \right\} (5)$ $\left. \begin{array}{l} /:D \\ /C:C \end{array} \right\} (1)$
141	1.000	0.0	$\left. \begin{array}{l} /:D \\ C/D:D \end{array} \right\} (8)$ $\left. \begin{array}{l} /:D \\ /C:D \end{array} \right\} (7)$ $\left. \begin{array}{l} /:D \\ /D:D \end{array} \right\} (2)$ $\left. \begin{array}{l} /:D \\ /D:D \end{array} \right\} (1)$ $\left. \begin{array}{l} /:D \\ D/D:D \end{array} \right\} (1)$
142	1.017	0.0	$\left. \begin{array}{l} /:D \\ /C:D \end{array} \right\} (7)$ $\left. \begin{array}{l} /:D \\ /D:D \end{array} \right\} (6)$ $\left. \begin{array}{l} /:D \\ C/D:D \end{array} \right\} (4)$ $\left. \begin{array}{l} /:D \\ /C:C \end{array} \right\} (2)$ $\left. \begin{array}{l} /:C \\ /D:D \end{array} \right\} (1)$
143	1.314	2.8	$\left. \begin{array}{l} /:D \\ /C:C \end{array} \right\} (6)$ $\left. \begin{array}{l} /:D \\ /D:D \end{array} \right\} (3)$ $\left. \begin{array}{l} /:D \\ C/D:D \end{array} \right\} (3)$ $\left. \begin{array}{l} /:D \\ /C:D \end{array} \right\} (3)$ $\left. \begin{array}{l} /:C \\ /D:D \end{array} \right\} (3)$ $\left. \begin{array}{l} /:D \\ C/C:D \end{array} \right\} (1)$ $\left. \begin{array}{l} /:C \\ /C:D \end{array} \right\} (1)$
144	2.365	43.7	$\left. \begin{array}{l} /:C \\ /D:D \end{array} \right\} (12)$ $\left. \begin{array}{l} /:D \\ /C:C \end{array} \right\} (5)$ $\left. \begin{array}{l} /:D \\ C/D:D \end{array} \right\} (2)$ $\left. \begin{array}{l} /:C \\ /C:C \end{array} \right\} (1)$

147	3.000	100.0	$\left. \begin{array}{l} /:C \\ /D:D \end{array} \right\} (20)$
274	3.000	100.0	$\left. \begin{array}{l} /:C \\ /C:C \end{array} \right\} (20)$
275	2.951	90.1	$\left. \begin{array}{l} /:C \\ /C:C \end{array} \right\} (19)$ $\left. \begin{array}{l} /:C \\ /C:D \end{array} \right\} (1)$
278	2.182	36.8	$\left. \begin{array}{l} /:C \\ /C:D \end{array} \right\} (16)$ $\left. \begin{array}{l} /:C \\ /C:C \end{array} \right\} (3)$ $\left. \begin{array}{l} /:C \\ /:D \end{array} \right\} (1)$
279	2.126	36.7	$\left. \begin{array}{l} /:C \\ /C:D \end{array} \right\} (15)$ $\left. \begin{array}{l} /:C \\ /:D \end{array} \right\} (3)$ $\left. \begin{array}{l} /:D \\ /C:D \end{array} \right\} (1)$
280	2.089	14.6	$\left. \begin{array}{l} /:C \\ /C:D \end{array} \right\} (6)$ $\left. \begin{array}{l} /:C \\ /:D \end{array} \right\} (6)$ $\left. \begin{array}{l} /:D \\ /C:D \end{array} \right\} (4)$ $\left. \begin{array}{l} /:D \\ /:D \end{array} \right\} (2)$ $\left. \begin{array}{l} /:C \\ /C:C \end{array} \right\} (1)$ $\left. \begin{array}{l} /:C \\ /C:C \end{array} \right\} (1)$
281	1.156	0.04	$\left. \begin{array}{l} /:D \\ /C:D \end{array} \right\} (14)$ $\left. \begin{array}{l} /:D \\ /:D \end{array} \right\} (2)$ $\left. \begin{array}{l} /:C \\ /:D \end{array} \right\} (2)$ $\left. \begin{array}{l} /:D \\ /C:C \end{array} \right\} (1)$ $\left. \begin{array}{l} /:D \\ C/:D \end{array} \right\} (1)$
282	1.000	0.0	$\left. \begin{array}{l} /:D \\ /C:D \end{array} \right\} (18)$ $\left. \begin{array}{l} /:D \\ /:D \end{array} \right\} (2)$

^a Population size = 20³; memory limit = 1; 101 interactions per pair; crossover probability = 0.21; mutation probabilities = 0.01.

in the IPD from defection towards cooperation. We tested this view in a series of EvA runs with different combinations of population size and interaction-sequence length.

Effects of population size and interaction-sequence length in the EvA model

The EvA results show a clear interaction between population size and interaction-sequence length (Fig. 4). For small populations or short sequences or both, only approximately 10% of interactions yield mutual cooperation. However, for the largest populations and longest sequences considered, the paired individuals both cooperated in more than half of all interactions. Direct examination of the underlying evolutionary sequences (Table 3) documents the replacement of ALL D by TFT and vice versa in ways completely consistent with the analysis of Appendix 2 and the interpretation in Fig. 3.

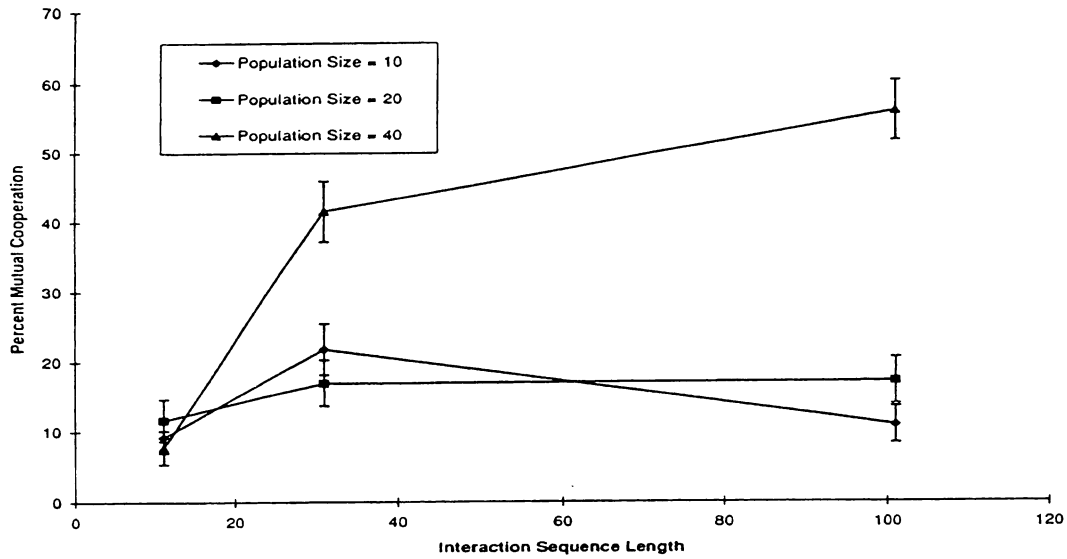
Discussion

To understand better the tit-for-tat strategy and its implications for the emergence of cooperation in interacting populations of selfish individuals, we have applied a newly developed evolutionary approach based on pairwise sequences of playing the prisoner's dilemma game. We have shown that TFT and a variety of other simple strategies can be represented in a format that facilitates observing and understanding their dynamics. We assumed that strategies (algorithms) are composed of two interacting components (rules), each of which can change incrementally. Analysis of these incremental changes (single-point mutations) indicated that TFT and ALL D are the only evolutionarily stable strategies in the absence of multiple simultaneous mutations and of genetic drift (cf. Axelrod and Hamilton, 1981).

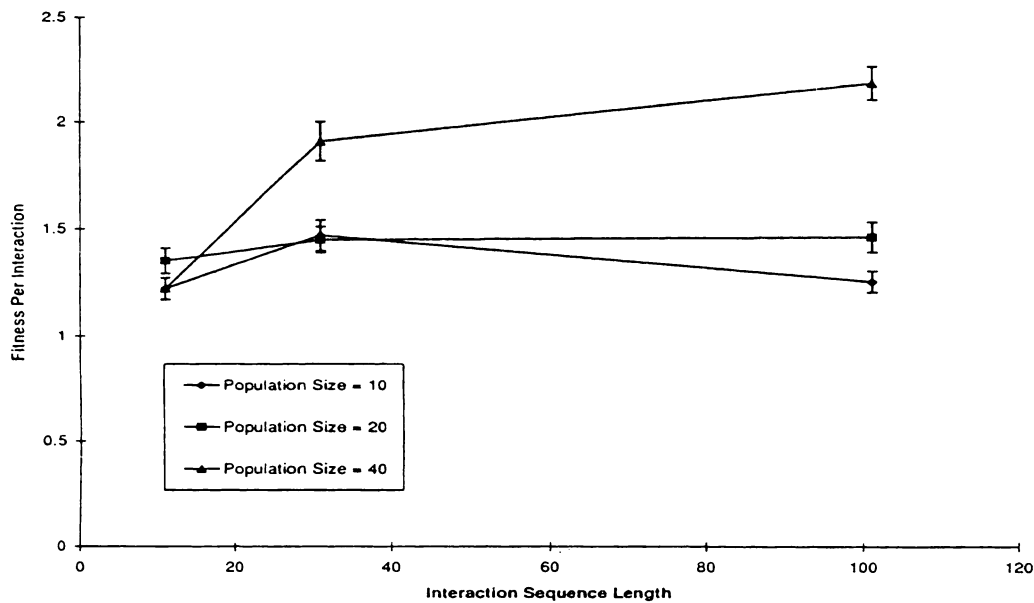
We then showed that drift and the accumulation of mutations under drift render TFT and ALL D each invulnerable to the other, though the dynamics of invasion and replacement are quite different in the two cases. This interpretation yielded the prediction that larger population size and longer interaction sequences would shift the balance of behaviour in the population from defection to cooperation and this was borne out in runs of a model explicitly depicting the evolutionary dynamics. Moreover, replacement sequences generated by the model were consistent with our analysis of the underlying mechanisms.

These results have been obtained from a particularly simple and restricted array of behavioural strategies for playing the iterated prisoner's dilemma game. When behavioural strategies can be based on memory of more previous interactions (longer rules) or on more complex behavioural algorithms (more rules), TFT can and does still evolve. Moreover, many of the other strategies capable of generating cooperation are clearly based on TFT, since they contain the two TFT rules, along with some others directing responses to specific situations that can fine-tune behaviour in various ways. Such strategies may evolve from TFT and are often particularly TFT-like early in interaction sequences, when the interaction history on which the subtler responses are based does not yet exist (see Crowley, 1995).

The 35 algorithms considered here represent only 13 distinct IPD strategies, since the 12 ALL C and the 12 ALL D algorithms are functionally indistinguishable (within each set of 12) in playing the IPD. It is important to realize, however, that there are no evolutionarily stable strategies even among this restricted set, except with fewer than three interactions per sequence. We note that this evolutionary transience clearly does not hinge on noise in interpreting or responding to another individual's behaviour (see Lindgren, 1991) but *does* depend on genetic drift, in which strategically equivalent strategies can vary randomly in frequency across generations. Of course, evolutionary transience does not preclude the long-term dominance of one or a few strategies, as implied by our



(a)



(b)

Figure 4. Mean percent mutual cooperation ± 1 SE (a) and mean fitness per round of the IPD game ± 1 SE (b) versus population size and number of rounds per IPD game. Note that percent mutual cooperation and fitness are tightly correlated, in that the two patterns are very similar. These data were obtained from the EvA model (P.H. Crowley, submitted) with two rules per algorithm, memory limit = 1 (100 replicates per mean), the results averaged over the final 500 of the 1000 generations, and mutation rate = 0.01 and crossover rate = 0.21 per locus per generation. The runs were initialized with random rules, approximately 44% of which were zero-order, 44% first-order and 11% second-order.

results with the EvA simulations. Moreover, selective forces may ultimately eliminate the transience of cooperative strategies. For example, once transient cooperation appears, some combination of individual, kin and group selection may adjust the amount of memory (Healy, 1992) or type of memory (Crowley *et al.*, 1995) used, the cognitive structure (Churchland and Sejnowski, 1992), deme structure (Wilson, 1979, 1980), IPD pay-offs (Axelrod and Hamilton, 1981) and other features to shift the balance sufficiently for cooperation to become stable. This possibility and these potential mechanisms deserve further attention.

Our analysis suggests some situations in which TFT might be important in nature, relative to more complex cooperative strategies.

(1) For cognitively simple organisms of limited memory capacity, the situation explicitly depicted here.

(2) In strongly kin- or group-selected populations. Though low memory capacity can actually yield higher fitness (Crowley, 1995; see also Healy, 1992), individuals with more memory and thus capable of more sophisticated strategies may eliminate low-memory individuals – except under strong kin or group selection (Crowley, 1995).

(3) When interaction sequences are relatively short – short enough that complex strategies are no more likely to be successful, but long enough that reciprocation can be important. The assumption in some recent studies that interaction sequences are infinitely long (e.g. Lindgren, 1991; Nowak and Sigmund, 1992, 1993) may severely limit the applicability of those analyses to many situations in nature.

(4) In moderate-sized populations of widely interacting individuals – or perhaps large populations composed of moderate-sized trait groups. Our analysis shows cooperative behaviour may be rapidly eroded by drift in small populations. However, in large populations, interacting with very many other individuals may mean that interaction sequences with each partner must be short, which (as we have seen) makes cooperation less likely to evolve. The role of tightly interacting subpopulations (trait groups) for evolving cooperation in large populations is currently being investigated (B. Spohn and P.H. Crowley, in preparation).

(5) When defectors appear in the population with moderate frequency. In this case, each defector reinoculates the population against defection without overwhelming the effects of selection favouring TFT. Under these conditions TFT's relative lack of generosity to defectors is an important asset (cf. Nowak and Sigmund, 1992).

Because the above situations should generally be common, we fully expect tit-for-tat and closely related strategies to be well represented in nature.

Acknowledgements

We thank Lee Dugatkin, Marcel Holyoak, Frank Johansson, Mark Kirkpatrick, Sharon Lawler and Gosta Nachman for helpful discussions and comments on the manuscript. We acknowledge support from NSF-Kentucky EPSCoR grant EHR-9108764.

References

- Axelrod, R. (1984) *The Evolution of Cooperation*. Basic Books, New York.
- Axelrod, R. (1987) The evolution of strategies in the iterated prisoner's dilemma. In *Genetic Algorithms and Simulated Annealing* (L. Davis, ed.), pp. 32–41. Pitman, London.
- Axelrod, R. and Dion, D. (1988) The further evolution of cooperation. *Science* **242**, 1385–90.
- Axelrod, R. and Hamilton, W.D. (1981) The evolution of cooperation. *Science* **211**, 1390–6.
- Boyd, R. and Lorberbaum, J.P. (1987) No pure strategy is evolutionarily stable in the repeated prisoner's dilemma game. *Nature* **327**, 58–9.

- Brown, J.L. (1987) *Helping and Communal Breeding in Birds*. Princeton University Press, Princeton, NJ.
- Churchland, P.S. and Sejnowski, T.J. (1992) *The Computational Brain*. MIT Press, Cambridge, MA.
- Crowley, P.H. (1995) Evolving cooperation: Strategies as hierarchies of rules. *BioSystems*, **37**, 67–80.
- Crowley, P.H., Provencher, L., Sloane, S., Dugatkin, L.A., Spohn, B., Rogers, B.L. and Alfieri, M. (1995) Evolving cooperation: The role of individual recognition. *BioSystems*, **37**, 49–66.
- Davis, L. (ed.) (1991) *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.
- Dugatkin, L.A. (1991) Dynamics of the TIT FOR TAT strategy during predator inspection in the guppy (*Poecilia reticulata*). *Behav. Ecol. Sociobiol.* **29**, 128–32.
- Dugatkin, L.A. and Alfieri, M. (1991) Guppies and the TIT FOR TAT strategy: preference based on past interaction. *Behav. Ecol. Sociobiol.* **28**, 243–6.
- Elster, J. (1979) *Ulysses and the Sirens*. Cambridge University Press, London.
- Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA.
- Hamilton, W.D. (1964) The genetical evolution of social behavior. *J. Theor. Biol.* **7**, 1–32.
- Harrison, S. (1991) Local extinction in a metapopulation context: an empirical evaluation. *Biol. J. Linn. Soc.* **42**, 73–88.
- Healy, S. (1992) Optimal memory: toward an evolutionary ecology of animal cognition? *Trends Evol. Ecol.* **7**, 399–400.
- Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI.
- Holland, J.H. (1992) Genetic algorithms. *Sci. Am.* **267**, 66–72.
- Lindgren, K. (1991) Evolutionary phenomena in simple dynamics. In *Artificial Life II, SFI Studies in the Science of Complexity*, Vol. X (C.G. Langton, C. Taylor, J.D. Farmer and S. Rasmussen, eds), pp. 295–312. Addison-Wesley, Reading, MA.
- Mesterton-Gibbons, M. (1992) On the iterated prisoner's dilemma in a finite population. *Bull. Math. Biol.* **54**, 423–43.
- Mesterton-Gibbons, M. and Dugatkin, L.A. (1992) Cooperation among unrelated individuals: evolutionary factors. *Q. Rev. Biol.* **67**, 267–81.
- Milinski, M. (1987) TIT FOR TAT and the evolution of cooperation in sticklebacks. *Nature* **325**, 433–5.
- Milinski, M. (1990) No alternative to TIT FOR TAT in sticklebacks. *Anim. Behav.* **39**, 989–91.
- Nowak, M.A. and Sigmund, K. (1992) Tit for tat in heterogeneous populations. *Nature* **355**, 250–3.
- Nowak, M.A. and Sigmund, K. (1993) A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature* **364**, 56–8.
- Sumida, B.H., Houston, A.I., McNamara, J.M. and Hamilton, W.D. (1990) Genetic algorithms and evolution. *J. Theor. Biol.* **147**, 59–84.
- Trivers, R.L. (1971) The evolution of reciprocal altruism. *Q. Rev. Biol.* **46**, 35–57.
- Trivers, R.L. (1985) *Social Evolution*. Benjamin/Cummings, Menlo Park, CA.
- West Eberhard, M.J. (1975) The evolution of social behavior by kin selection. *Q. Rev. Biol.* **50**, 1–33.
- Williams, G.C. (1966) *Adaptation and Natural Selection*. Princeton University Press, Princeton, NJ.
- Wilson, D.S. (1979) Structured demes and trait-group variation. *Am. Nat.* **113**, 606–10.
- Wilson, D.S. (1980) *The Natural Selection of Populations and Communities*. Benjamin/Cummings, Menlo Park, CA.
- Wright, S. (1921) Systems of mating. *Genetics* **6**, 111–78.

Appendix 1: determining the fate of single-point mutants of tit-for-tat and other algorithms

Table 1 summarizes the expected fate of all single-point mutants of the 35 two-rule algorithms. Here, we show how these conclusions were drawn, focusing mainly on the non-drift transitions of tit-for-tat (TFT) and suspicious tit-for-tat (STFT). For each algorithm of interest, we show enough of the relevant interaction sequences from the beginning to identify the repeat pattern, a short

sequence of responses by the two algorithms that is repeated throughout the remainder of the interaction. (In the cases below, the repeat patterns are enclosed in rectangles.) The mean pay-off per interaction for each algorithm in each iteration of the repeat pattern, determined using the values in the matrix of Fig. 1, then expresses the overall fitnesses of the algorithms. However, other subtle yet important fitness differences between algorithms can arise in two ways: (1) interactions preceding the first iteration of the repeat pattern for which the algorithms differ in pay-off (see case 1 below) and (2) ending the full sequence with a truncated repeat pattern (see case 2 below). In these situations, the mean pay-off per interaction may be altered by a small amount that depends on the total length of the interaction sequence. We indicate these effects as plusses and minuses beside the mean pay-off per interaction.

For algorithms A and B, the three relevant interaction sequences are A versus B, A versus A and B versus B. For each sequence, we find the repeat pattern, determine the expected pay-off for each algorithm and indicate plus or minus as appropriate. We conclude (1) that A can invade B if A does better against B than B does against itself, (2) that A and B drift if the algorithms do equally well in all interactions or (3) that A cannot invade B if A does worse against B than B does against itself. When each can invade the other, the result is a stable mixture; for sufficiently long interaction sequences, the proportions of the two algorithms in the mixture are determined by expressing the fitnesses of the two algorithms as functions of the proportions, equating the fitnesses and solving for the proportions.

Case 1: TFT versus probabilistic algorithm 2 in Table 1

$$2 = \begin{cases} /:C \\ /:D \end{cases} \quad 10 = \begin{cases} /:C \\ /D:D \end{cases}$$

1	2	3	←	interaction sequence
2	(C,D)	(C,D)	...	2.25 +
10	C	(C,D)	(C,D)	...

... 2.25 - where (C,D) implies C and D are equally probable

2	(C,D)	(C,D)	...	}	2.25
2	(C,D)	(C,D)	...		

10	C	C	...	}	3.0
10	C	C	...		

2 → 10	=	2.25 +	}	⇒ 2 cannot invade 10	10 → 2	=	2.25 -	}	⇒ 10 cannot invade 2
10 → 10	=	3.0			2 → 2	=	2.25		

Case 2: TFT versus algorithm 8, which defects following cooperation by its partner

$$8 = \begin{cases} /:C \\ /C:D \end{cases} \quad 10 = \begin{cases} /:C \\ /D:D \end{cases}$$

	1	2	3	4	5		
8	C	D	D	C	C	2.25 +
10	C	C	D	D	C	2.25 -

8	C	D	C	}	2.0 +	10	C	C	}	3.0
8	C	D	C			10	C	C		

$$\left. \begin{array}{l} 8 \rightarrow 10 = 2.25 + \\ 10 \rightarrow 10 = 3.0 \end{array} \right\} \Rightarrow 8 \text{ cannot invade } 10 \quad \left. \begin{array}{l} 10 \rightarrow 8 = 2.25 - \\ 8 \rightarrow 8 = 2.0 + \end{array} \right\} \Rightarrow 10 \text{ can invade } 8$$

Case 3: TFT versus ALL D algorithm 27, a key player in the evolution of TFT

$$10 = \begin{cases} /:C \\ /D:D \end{cases} \quad 27 = \begin{cases} /:D \\ /D:D \end{cases}$$

	1	2	3		
10	C	D	D	1.0 -
27	D	D	D	1.0 +

27	D	D	}	1.0	10	C	C	}	3.0
27	D	D			10	C	C		

$$\left. \begin{array}{l} 10 \rightarrow 27 = 1.0 - \\ 27 \rightarrow 27 = 1.0 \end{array} \right\} \Rightarrow 10 \text{ cannot invade } 27 \quad \left. \begin{array}{l} 27 \rightarrow 10 = 1.0 + \\ 10 \rightarrow 10 = 3.0 \end{array} \right\} \Rightarrow 27 \text{ cannot invade } 10 \text{ (for sequences longer than 2)}$$

Note that against TFT, the other three single-point mutants of TFT (algorithms 9, 14 and 18) always cooperate, as does TFT, resulting in equal pay-offs and genetic drift between TFT and any of the three.

Case 4: STFT versus probabilistic algorithm 2

$$2 = \begin{cases} /:C \\ /:D \end{cases} \quad 24 = \begin{cases} /:D \\ /C:C \end{cases}$$

	1	2	3		
2	(C,D)	(C,D)	(C,D)	2.25 -
24	D	(C,D)	(C,D)	2.25 +

2	(C,D)	(C,D)	}	= 2.25	24	D	D	}	1.0
2	(C,D)	(C,D)			24	D	D		

$$\left. \begin{array}{l} 2 \rightarrow 24 = 2.25 - \\ 24 \rightarrow 24 = 1.0 \end{array} \right\} \Rightarrow 2 \text{ can invade } 24 \quad \left. \begin{array}{l} 24 \rightarrow 2 = 2.25 + \\ 2 \rightarrow 2 = 2.25 \end{array} \right\} \Rightarrow 24 \text{ can invade } 2$$

Since algorithm 2's fitness advantage in invading algorithm 24 exceeds algorithm 24's fitness advantage in invading algorithm 2, the result is a stable mixture, with more algorithm 2 than algorithm 24. The expected proportion of algorithm 24 decreases with the length of the interaction sequence and asymptotically approaches 0.

Case 5: STFT versus ALL C algorithm 7

$$7 = \begin{cases} /:C \\ /C:C \end{cases} \quad 24 = \begin{cases} /:D \\ /C:C \end{cases}$$

	1	2	3		
7	C	C	C	3.0 -
24	D	C	C	3.0 +

	1	2			
7	C	C	}	3.0
7	C	C		
	1	2			
24	D	D	}	1.0
24	D	D		

$$\left. \begin{array}{l} 7 \rightarrow 24 = 3.0 - \\ 24 \rightarrow 24 = 1.0 \end{array} \right\} \Rightarrow 7 \text{ can invade } 24 \quad \left. \begin{array}{l} 24 \rightarrow 7 = 3.0 + \\ 7 \rightarrow 7 = 3.0 \end{array} \right\} \Rightarrow 24 \text{ can invade } 7$$

Since algorithm 7's fitness advantage in invading algorithm 24 exceeds algorithm 24's fitness advantage in invading algorithm 7, the result is a stable mixture, with more algorithm 7 than algorithm 24. The proportion of algorithm 24 decreases as interaction sequences lengthen and asymptotically approaches 0.

Case 6: STFT versus algorithm 26, which cooperates only after defection by its partner

$$24 = \begin{cases} /:D \\ /C:C \end{cases} \quad 26 = \begin{cases} /:D \\ /D:C \end{cases}$$

	1	2	3	4	5	
24	D	D	C	C	D 2.25 +
26	D	C	C	D	D 2.25 -

	1	2			
24	D	D	}	1.0
24	D	D		
	1	2	3		
26	D	C	D	}
26	D	C	D	

$$\left. \begin{array}{l} 24 \rightarrow 26 = 2.25 + \\ 26 \rightarrow 26 = 2.0 - \end{array} \right\} \Rightarrow 24 \text{ can invade } 26 \quad \left. \begin{array}{l} 26 \rightarrow 24 = 2.25 - \\ 24 \rightarrow 24 = 1.0 \end{array} \right\} \Rightarrow 26 \text{ can invade } 24$$

Since algorithm 26's fitness advantage in invading algorithm 24 exceeds algorithm 24's fitness advantage in invading algorithm 26, the result is a stable mixture, with more algorithm 26 than algorithm 24. Let p be the proportion of algorithm 24 in the mixture of algorithms 24 and 26. Then the fitness of algorithm 24 is $F_{24} = p + 2.25(1 - p)$ and the fitness of algorithm 26 is $F_{26} = 2.25p + 2.0(1 - p)$. Setting $F_{24} = F_{26}$ then yields $p = 1/6$. The proportion of algorithm 24 decreases as interaction sequences lengthen and asymptotically approaches 1/6.

Against STFT, the other three single-point mutants of STFT (algorithms 25, 28 and 32) always defect, along with STFT, producing equal pay-offs and genetic drift between STFT and any of these.

Case 7: algorithm 15, the only ALL C algorithm that is eliminated by none of its single-point mutants, versus algorithm 32, the initial-defection version of 15

$$15 = \begin{cases} /:C \\ D/C:C \end{cases} \quad 32 = \begin{cases} /:D \\ D/C:C \end{cases}$$

	1	2	3		
15	C	C	C	1.5 -
32	D	C	D	4.0 +

15	C	C	}	3.0	32	D	D	}	1.0
15	C	C			32	D	D		

$$\left. \begin{array}{l} 15 \rightarrow 32 = 1.5 - \\ 32 \rightarrow 32 = 1.0 \end{array} \right\} \Rightarrow 15 \text{ can invade } 32 \quad \left. \begin{array}{l} 32 \rightarrow 15 = 4.0 + \\ 15 \rightarrow 15 = 3.0 \end{array} \right\} \Rightarrow 32 \text{ can invade } 15$$

Let p be the proportion of algorithm 15 in the stable mixture when interaction sequences are very long (or even-numbered) and let F_{15} and F_{32} be the fitnesses of the two strategies. Then $F_{15} = 3.0p + 1.5(1 - p)$ and $F_{32} = 4.0p + (1 - p)$. Then $F_{15} = F_{32}$ implies that $p = 1/3$. Thus, the result is a stable mixture of strategies 15 and 32; algorithm 15 is increasingly represented for longer (or even-numbered) interaction sequences, with an upper limit of $p = 1/3$.

These examples illustrate the methods by which any of the other results summarized in Table 1 can readily be obtained.

Appendix 2: how do TFT and ALL D manage to invade each other?

The invasion of ALL D by TFT will be considered first in this analysis and then the reverse. An invasion by strategy A will be considered successful here if a single mutant A individual can achieve a greater total pay-off against its opponents than the average opponent achieves. The population consists of the one mutant and n other individuals. All interaction sequences are of length m , which is assumed to be an odd number, as in the EvA runs described in the text. (The relationships below can readily be modified to represent the cases with m even or with m equally likely to be even or odd; for these cases, the inequalities presented here are only very slightly conservative.) The 12 ALL D algorithms of Table 1 (those on the right-hand side having their numbers within a square) are equivalent for the present purpose and will be referred to collectively in this appendix as ALL D. Similarly, the 12 ALL C algorithms in the table will be collectively labelled ALL C.

Invasion of ALL D by TFT

TFT can invade ALL D only when there are enough other individuals present that will cooperate in response to cooperation, overcompensating TFT for initially playing C against defectors. Of the 35 two-rule algorithms, only STFT can drift into a population of ALL D carrying this critical capability.

Let s be the number of STFT individuals among the n defectors (STFT + ALL D). Pay-offs (via Fig. 1) over all opponents, at m rounds per opponent, are as follows.

$$\text{TFT: } 2.5s(m-1) + (n-s)(m-1)$$

$$\text{ALL D: } mn + 4$$

$$\text{STFT: } m(n-1) + 2.5(m+1)$$

$$\text{Average opponent of TFT: } ((n-s)/n)(mn+4) + (s/n)(m(n-1) + 2.5(m+1))$$

Successful invasion thus implies

$$2.5s(m-1)n + (n-s)(m-1)n > (n-s)(mn+4) + s(m(n-1) + 2.5(m+1))$$

which implies

$$s[1.5n(m-1) + mn + 4 - m(n-1) - 2.5(m+1)] > n(mn+4) - n^2(m-1)$$

resulting in

$$s > \frac{n(n+4)}{1.5(m-1)(n-1)}$$

Let s^* be the minimal s consistent with successful invasion. Then for very long interaction sequences, $s^* \rightarrow 0$. For very large populations, the frequency s^*/n approaches $1/(1.5m - 1.5)$.

Invasion of TFT by ALL D

ALL D can invade TFT only in the presence of a substantial number of ALL C individuals, which can drift into a population of TFT. Let c be the number of ALL C individuals among the n cooperators (TFT + ALL C). Pay-offs over all rounds and opponents are as follows.

$$\text{ALL D: } 5cm + (n-c)(m+4)$$

$$\text{TFT: } 3m(n-1) + m-1$$

$$\text{ALL C: } 3m(n-1)$$

$$\text{Average opponent of ALL D: } 3m(n-1) + ((n-c)/n)(m-1)$$

The invasion succeeds when

$$5cmn + n(n-c)(m+4) > 3mn(n-1) + (n-c)(m-1)$$

which implies

$$c(5mn - (m+4)n + m-1) > 3mn(n-1) + n(m-1) - (m+4)n^2$$

yielding

$$c > \frac{n(2m(n-1) - 4n - 1)}{(4n+1)(m-1)}$$

Let c^* be the minimal number of ALL C individuals required for the invasion to succeed. Then for very long sequences, the frequency $c^*/n = 2(n-1)/(4n+1)$; for very large populations, $c^*/n = (2m-4)/(4m-4)$; and when both m and n are large, $c^*/n = 1/2$.

Thus, with the exception of when populations are very small and sequences are short, approximately half of a TFT population must have drifted to ALL C for an ALL D mutant to invade successfully, whereas very few (often only one) STFT are needed to make an ALL D population invulnerable to TFT.