Classification of Dilemmas
## General Payoff Matrix

<table>
<thead>
<tr>
<th></th>
<th>Ann</th>
<th>Bob</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>cooperate</td>
</tr>
<tr>
<td>cooperate</td>
<td></td>
<td>CC (R)</td>
</tr>
<tr>
<td>defect</td>
<td></td>
<td>DC (T)</td>
</tr>
</tbody>
</table>
General Conditions for a Dilemma

- You always benefit if the other cooperates:
  - $CC > CD \text{ and } DC > DD$
- You sometimes benefit from defecting:
  - $DC > CC \text{ or } DD > CD$
- Mutual coop. is preferable to mut. def.:
  - $CC > DD$
- Consider relative size of $CC, CD, DC, DD$
  - think of as permutations of $R, S, T, P$
  - only three result in dilemmas
Three Possible Orders

The three dilemmas: TRSP, RTPS, TRPS
The Three Dilemmas

- **Chicken** (*TRSP*)
  - DC > CC > CD > DD
  - characterized by mutual defection being worst

- **Stag Hunt** (*RTPS*)
  - CC > DC > DD > CD
  - better to cooperate with cooperator

- **Prisoners’ Dilemma** (*TRPS*)
  - DC > CC > DD > CD
  - better to defect on cooperator
The Iterated Prisoners’ Dilemma

and Robert Axelrod’s Experiments
Assumptions

• No mechanism for enforceable threats or commitments
• No way to foresee a player’s move
• No way to eliminate other player or avoid interaction
• No way to change other player’s payoffs
• Communication only through direct interaction
Axelrod’s Experiments

• Intuitively, expectation of future encounters may affect rationality of defection
• Various programs compete for 200 rounds
  – encounters each other and self
• Each program can remember:
  – its own past actions
  – its competitors’ past actions
• 14 programs submitted for first experiment
# IPD Payoff Matrix

<table>
<thead>
<tr>
<th></th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cooperate</td>
</tr>
<tr>
<td>cooperate</td>
<td>3, 3</td>
</tr>
<tr>
<td>defect</td>
<td>5, 0</td>
</tr>
</tbody>
</table>

N.B. Unless DC + CD < 2 CC (i.e. $T + S < 2 R$), can win by alternating defection/cooperation
Indefinite Number of Future Encounters

• Cooperation depends on expectation of **indefinite** number of future encounters

• Suppose a known finite number of encounters:
  – No reason to C on last encounter
  – Since expect D on last, no reason to C on next to last
  – And so forth: there is no reason to C at all
Analysis of Some Simple Strategies

• Three simple strategies:
  – **ALL-D**: always defect
  – **ALL-C**: always cooperate
  – **RAND**: randomly cooperate/defect

• Effectiveness depends on environment
  – **ALL-D** optimizes local (individual) fitness
  – **ALL-C** optimizes global (population) fitness
  – **RAND** compromises
### Expected Scores

<table>
<thead>
<tr>
<th>Playing</th>
<th>ALL-C</th>
<th>RAND</th>
<th>ALL-D</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL-C</td>
<td>3.0</td>
<td>1.5</td>
<td>0.0</td>
<td>1.5</td>
</tr>
<tr>
<td>RAND</td>
<td>4.0</td>
<td>2.0</td>
<td>0.5</td>
<td>2.166…</td>
</tr>
<tr>
<td>ALL-D</td>
<td>5.0</td>
<td>3.0</td>
<td>1.0</td>
<td>3.0</td>
</tr>
</tbody>
</table>
Result of Axelrod’s Experiments

- **Winner** is Rapoport’s **TFT** (Tit-for-Tat)
  - cooperate on first encounter
  - reply in kind on succeeding encounters
- **Second experiment:**
  - 62 programs
  - all know **TFT** was previous winner
  - **TFT** wins again
Characteristics of Successful Strategies

• *Don’t be envious*
  – at best TFT ties other strategies
• *Be nice*
  – i.e. don’t be first to defect
• *Reciprocate*
  – reward cooperation, punish defection
• *Don’t be too clever*
  – sophisticated strategies may be unpredictable & look random