What Coalitions Can Achieve

Jürgen Dix and Wojtek Jamroga

Department of Informatics
Clausthal University of Technology, Germany

European Agent Systems Summer School (Lisbon 2008)
Lecture Overview

1. Introduction
2. How to Form a Coalition
3. Reasoning about Coalitions
Pointers to Other Courses

- Paul Harrenstein/ Mathijs de Weerdt: *Introduction to Game Theory and Mechanism Design* (Tuesday 11-16)
Timetable

- Section 1: Required Concepts.  
  20-30 min. Very brief recap.

- Section 2: How to form a coalition.  
  60-70 min: Standard.

- Section 3: Reasoning about Coalitions.  
  90 min: Advanced.

Updated slides: http://cig.in.tu-clausthal.de/index.php\relax?id=159
1. Required Concepts

Chapter 1. Required Concepts

Required Concepts
1.1 Evaluation Criteria
1.2 Non-Coop Games in NF
1.3 Non-Coop Extensive Games
1.4 References
We consider in this chapter non-cooperative games, state several evaluation criteria, the minmax theorem and its famous generalization: Nash’s theorem, introduce two different sorts of games: those in normal form and those in extensive form.
1. Required Concepts

**Classical DAI:** System Designer fixes an **Interaction-Protocol** which is uniform for all agents. The **designer also fixes a strategy for each agent.**

What is the outcome, assuming that the **protocol** is followed and the agents follow the strategies?
1. Required Concepts

**MAS:** Interaction-Protocol is given. Each agent determines its own strategy (maximising its own good, via a utility function, without looking at the global task).

**Global optimum**

What is the outcome, given a protocol that guarantees that each agent’s desired local strategy is the best one (and is therefore chosen by the agent)?
1.1 Evaluation Criteria
We need to **compare protocols**. Each such protocol leads to a solution. So we determine how good these solutions are.

**Social Welfare:** Sum of all utilities

**Pareto Efficiency:** A solution $x$ is Pareto-optimal, if there is no solution $x'$ with:

1. $\exists$ agent $ag : ut_{ag}(x') > ut_{ag}(x)$
2. $\forall$ agents $ag' : ut_{ag'}(x') \geq ut_{ag'}(x)$.

**Individual rational:** if the payoff is higher than not participating at all.
Stability:

Case 1: Strategy of an agent depends on the others. The profile $S^*_A = \langle S^*_1, S^*_2, \ldots, S^*_{|A|} \rangle$ is called a Nash-equilibrium, iff $\forall i : S^*_i$ is the best strategy for agent $i$ if all the others choose $\langle S^*_1, S^*_2, \ldots, S^*_i-1, S^*_i+1, \ldots, S^*_{|A|} \rangle$.

Case 2: Strategy of an agent does not depend on the others. Such strategies are called dominant.
Example 1.1 (Prisoners Dilemma, Type 1)

Two prisoners are suspected of a crime (which they both committed). They can choose to (1) cooperate with each other (not confessing to the crime) or (2) defect (giving evidence that the other was involved). Both cooperating (not confessing) gives them a shorter prison term than both defecting. But if only one of them defects (the betrayer), the other gets maximal prison term. The betrayer then has maximal payoff.

<table>
<thead>
<tr>
<th>Prisoner 1</th>
<th>Prisoner 2</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>
1. Required Concepts

- **Social Welfare**: Both cooperate,
- **Pareto-Efficiency**: All are Pareto optimal, except when both defect.
- **Dominant Strategy**: Both defect.
- **Nash Equilibrium**: Both defect.
1.2 Non-Coop Games in NF
Prisoners dilemma revisited: $c \succeq a \succeq d \succeq b$

<table>
<thead>
<tr>
<th>Prisoner 1</th>
<th>cooperate</th>
<th>defect</th>
</tr>
</thead>
<tbody>
<tr>
<td>cooperate</td>
<td>(a,a)</td>
<td>(b,c)</td>
</tr>
<tr>
<td>defect</td>
<td>(c,b)</td>
<td>(d,d)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Prisoner 2</th>
<th>cooperate</th>
<th>defect</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a,a)</td>
<td>(c,b)</td>
<td>(d,d)</td>
</tr>
</tbody>
</table>
### Example 1.2 (Trivial mixed-motive game, Type 0)

<table>
<thead>
<tr>
<th>Player 1</th>
<th>Player 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C</td>
</tr>
<tr>
<td>C</td>
<td>(4, 4)</td>
</tr>
<tr>
<td>D</td>
<td>(3, 2)</td>
</tr>
</tbody>
</table>
Example 1.3 (Battle of the Bismarck Sea)

In 1943 the northern half of New Guinea was controlled by the Japanese, the southern half by the allies. The Japanese wanted to reinforce their troops. This could happen using two different routes: (1) north (rain and bad visibility) or (2) south (weather ok). Trip should take 3 days.

The allies want to bomb the convoy as long as possible. If they search north, they can bomb 2 days (independently of the route taken by the Japanese). If they go south, they can bomb 3 days if the Japanese go south too, and only 1 day, if the Japanese go north.
### 1. Required Concepts

### 2. Non-Coop Games in NF

<table>
<thead>
<tr>
<th></th>
<th>Japanese</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sail North</td>
<td>Sail South</td>
</tr>
<tr>
<td><strong>Allies</strong></td>
<td><strong>Search North</strong></td>
<td>2 days</td>
</tr>
<tr>
<td></td>
<td><strong>Search South</strong></td>
<td>2 days</td>
</tr>
<tr>
<td></td>
<td>1 day</td>
<td>3 days</td>
</tr>
</tbody>
</table>

**Allies**: *What is the largest of all row minima?*

**Japanese**: *What is smallest of the column maxima?*

**Battle of the Bismarck sea:**

*largest row minimum = smallest column maximum.*

This is called a **saddle point**.
### Example 1.4 (Rochambeau Game)

Also known as paper, rock and scissors: paper covers rock, rock smashes scissors, scissors cut paper.

<table>
<thead>
<tr>
<th></th>
<th>Min</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>S</td>
<td>R</td>
</tr>
<tr>
<td>Max</td>
<td>P</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Definition 1.5 (\(n\)-Person Normal Form Game)

A finite \(n\)-person normal form game is a tuple \(\langle A, A, O, \varrho, \mu \rangle\), where

- \(A = \{1, \ldots, i, \ldots, n\}\) is a finite set of players.
- \(A = \prod_1^n A_i\), where \(A_i\) is the set of actions available to player \(i\). A vector \(a \in A\) is called action profile. The elements of \(A_i\) are also called pure strategies.
- \(O\) is the set of outcomes.
- \(\varrho : A \rightarrow O\) assigns each action profile an outcome.
- \(\mu = \langle \mu_1, \ldots, \mu_i, \ldots, \mu_n \rangle\) where \(\mu_i : O \rightarrow \mathbb{R}\) is a real-valued utility (payoff) function for player \(i\).
Games can be represented graphically using an \(n\)-dimensional payoff matrix. Here is a generic picture for 2-player, 2-strategy games:

<table>
<thead>
<tr>
<th></th>
<th>(a_1)</th>
<th>(a_2)</th>
<th>(a_1)</th>
<th>(a_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_1)</td>
<td>(\mu_1(a_1^1, a_1^2), \mu_2(a_1^1, a_1^2))</td>
<td>(\mu_1(a_1^1, a_1^2), \mu_2(a_1^1, a_2^2))</td>
<td>(\mu_1(a_1^2, a_1^2), \mu_2(a_1^2, a_1^2))</td>
<td>(\mu_1(a_2^2, a_2^2), \mu_2(a_2^2, a_2^2))</td>
</tr>
<tr>
<td>(a_2)</td>
<td>(\mu_1(a_2^1, a_2^2), \mu_2(a_2^1, a_2^2))</td>
<td>(\mu_1(a_2^1, a_2^1), \mu_2(a_2^1, a_2^1))</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We often forget about \(\varrho\) (thus we are making no distinction between actions and outcomes). Thus we simply write \(\mu_1(a_1^1, a_1^1)\) instead of the more precise \(\mu_1(\varrho(\langle a_1^1, a_1^2 \rangle))\).
Definition 1.6 (Common Payoff Game)

A **common payoff game (team game)** is a game in which for all action profiles \( a \in A_1 \times \ldots \times A_n \) and any two agents \( i, j \) the following holds:
\[
\mu_i(a) = \mu_j(a).
\]

In such games agents have no conflicting interests. Their graphical depiction is simpler than above (the second component is not needed).
The opposite of a team game is a

**Definition 1.7 (Constant Sum Game)**

A 2-player \( n \)-strategy normal form game is called **constant sum game**, if there exists a constant \( c \) such that for each strategy profile \( a \in A_1 \times A_2 \):

\[
\mu_1(a) + \mu_2(a) = c.
\]

We usually set wlog \( c = 0 \) (zero sum games).

What we are really after is **strategies**. A pure **strategy** is one where one action is chosen and played. But does this always make sense?
Example 1.8 (Battle of the Sexes, Type 2)

Married couple looks for evening entertainment. They prefer to go out together, but have different views about what to do (say going to the theatre and eating in a gourmet restaurant).

<table>
<thead>
<tr>
<th>Husband</th>
<th>Wife</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theatre</td>
<td>Theatre</td>
</tr>
<tr>
<td>Restaurant</td>
<td>Restaurant</td>
</tr>
<tr>
<td>(1, 1)</td>
<td>(3, 4)</td>
</tr>
</tbody>
</table>
Example 1.9 (Leader Game, Type 3)

Two drivers attempt to enter a busy stream of traffic. When the cross traffic clears, each one has to decide whether to concede the right of way of the other (C) or drive into the gap (D). If both decide for C, they are delayed. If both decide for D there may be a collision.

<table>
<thead>
<tr>
<th>Driver 1</th>
<th>Driver 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>(2, 2)</td>
</tr>
<tr>
<td>D</td>
<td>(3, 4)</td>
</tr>
<tr>
<td></td>
<td>(4, 3)</td>
</tr>
<tr>
<td></td>
<td>(1, 1)</td>
</tr>
</tbody>
</table>
Example 1.10 (Fighters and Bombers)

Consider fighter pilots in WW II. A good strategy to attack bombers is to swoop down from the sun: Hun-in-the-sun strategy. But the bomber pilots can put on their sunglasses and stare into the sun to watch the fighters. So another strategy is to attack them from below Ezak-Imak strategy: if they are not spotted, it is fine, if they are, it is fatal for them (they are much slower when climbing). The table contains the survival probabilities of the fighter pilot.

<table>
<thead>
<tr>
<th>Fighter Pilots</th>
<th>Bomber Crew</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Look Up</td>
</tr>
<tr>
<td>Hun-in-the-Sun</td>
<td>0.95</td>
</tr>
<tr>
<td>Ezak-Imak</td>
<td>1</td>
</tr>
</tbody>
</table>
Example 1.11 (Matching Pennies Game)

Two players display one side of a penny (head or tails). Player 1 wins the penny if they display the same, player 2 wins otherwise.

<table>
<thead>
<tr>
<th>Player 1</th>
<th>Player 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head</td>
<td>Head</td>
</tr>
<tr>
<td>Tails</td>
<td>Tails</td>
</tr>
<tr>
<td>(1, -1)</td>
<td>(-1, 1)</td>
</tr>
</tbody>
</table>

Jürgen Dix and Wojtek Jamroga · European Agent Systems Summer School (Lisbon 2008)
Definition 1.12 (Mixed Strategy for NF Games)

Let \( \langle A, A, O, \varrho, u \rangle \) be normal form game. For a set \( X \) let \( \prod(X) \) be the set of all probability distributions over \( X \). The set of mixed strategies for player \( i \) is the set \( S_i = \prod(A_i) \). The set of mixed strategy profiles is \( S_1 \times \ldots \times S_n \).

The **support** of a mixed strategy is the set of actions that are assigned non-zero probabilities.

What is the payoff of such strategies? We have to take into account the probability with which an action is chosen. This leads to the expected utility \( \mu^{expected} \).
Definition 1.13 (Expected Utility for player $i$)

The expected utility for player $i$ of the mixed strategy profile $(s_1, \ldots, s_n)$ is defined as

$$\mu_{\text{expected}}(s_1, \ldots, s_n) = \sum_{a \in A} \mu_i(\varrho(a)) \prod_{j=1}^{n} s_j(a_j).$$

What is the optimal strategy (maximising the expected payoff) for an agent in an 2-agent setting?
Definition 1.14 (Maxmin strategy)

Given a game \( \langle \{1, 2\}, \{A_1, A_2\}, \{\mu_1, \mu_2\} \rangle \), the maxmin strategy of player \( i \) is a mixed strategy that maximises the guaranteed payoff of player \( i \), no matter what the other player \( -i \) does:

\[
\arg\max_{s_i} \min_{s_{-i}} \mu_i^{\text{expected}}(s_1, s_2)
\]

The maxmin value for player \( i \) is

\[
\max_{s_i} \min_{s_{-i}} \mu_i^{\text{expected}}(s_1, s_2).
\]

The minmax strategy for player \( i \) is

\[
\arg\min_{s_i} \max_{s_{-i}} \mu_{-i}^{\text{expected}}(s_1, s_2)
\]

and its minmax value is

\[
\min_{s_{-i}} \max_{s_i} \mu_{-i}^{\text{expected}}(s_1, s_2).
\]
Lemma 1.15

In each finite normal form 2-person game (not necessarily constant sum), the maxmin value of one player is never strictly greater than the minmax value for the other.
We illustrate the maxmin strategy using a 2-person 3-strategy constant sum game:

<table>
<thead>
<tr>
<th>Player A</th>
<th>Player B</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B-I</td>
</tr>
<tr>
<td>A-I</td>
<td>0</td>
</tr>
<tr>
<td>A-II</td>
<td>1</td>
</tr>
</tbody>
</table>

We assume Player A’s optimal strategy is to play strategy

- A-I with probability $x$ and
- A-II with probability $1 - x$.

In the following we want to determine $x$. 
In accordance with the minmax strategy let us compute

\[ \text{argmax}_{s_i} \ \min_{s_{-i}} \ \mu_i^{\text{expected}}(s_1, s_2) \]

We assume Player A plays (as above) A-I with probability \( x \) and A-II with probability \( 1 - x \) (strategy \( s_1 \)). Similarly, Player B plays B-I with probability \( y \) and B-II with probability \( 1 - y \) (strategy \( s_2 \)).

We compute \( \mu_1^{\text{expected}}(s_1, s_2) \)

\[
0 \cdot x \cdot y + \frac{5}{6} x (1 - y) + 1 \cdot (1 - x) y + \frac{1}{2} (1 - x)(1 - y)
\]

thus

\[
\mu_1^{\text{expected}}(s_1, s_2) = y\left(\left(-\frac{4}{3} x\right) + \frac{1}{2}\right) + \frac{1}{3} x + \frac{1}{2}
\]
According to the minmax strategy, we have to choose $x$ such that the minimal values of the above term are maximal. For each value of $x$ the above is a straight line with some gradient. Thus we get the maximum when the line does not slope at all!

Thus $x = \frac{3}{8}$. A similar reasoning gives $y = \frac{1}{4}$. 
1. Required Concepts

2. Non-Coop Games in NF

Theorem 1.16 (von Neumann (1928))

In any finite 2-person constant-sum game the following holds:

1. The **maxmin value** for one player is equal to the **minmax value** for the other. The maxmin of player 1 is usually called **value of the game**.

2. For each player, the set of maxmin strategies coincides with the set of minmax strategies.

3. The maxmin strategies are **optimal**: if one player does not play a maxmin strategy, then its payoff (expected utility) goes down.

What is the optimal strategy (maximising the expected payoff) for an agent in an **$n$-agent setting**?
1. Required Concepts
2. Non-Coop Games in NF

Figure 1: A saddle.
Definition 1.17 (Best Response to a Strategy Profile)

Given a strategy profile

\[ \langle s_1, s_2, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n \rangle, \]

the **best response of player** \( i \) **to it** is any mixed strategy \( s_i^* \in S_i \) such that

\[
\mu_i(s_i^*, \langle s_1, s_2, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n \rangle) \geq \\
\mu_i(s_i, \langle s_1, s_2, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n \rangle)
\]

for all strategies \( s_i \in S_i \).

Is the best response **unique**?
Example 1.18 (Best responses sets for Rochambeau)

How does the set of best responses look like?

1. Player 2 plays the pure strategy P.
2. Player 2 plays P with probability \( \frac{1}{3} \) and S with probability \( \frac{2}{3} \).
3. Player 2 plays P with probability \( \frac{2}{3} \) and S with probability \( \frac{1}{3} \) and R with probability \( \frac{1}{3} \).

Is a non-pure strategy in the best response set (say a strategy \((s_1, s_2)\) with probabilities \(\langle p, 1 - p \rangle\), \(p \neq 0\)), then so are all other mixed strategies with probabilities \(\langle p', 1 - p' \rangle\) where \(p \neq p' \neq 0\).
1. Required Concepts

2. Non-Coop Games in NF

**Definition 1.19 (Nash-Equilibrium)**

A strategy profile $\langle s_1^*, s_2^*, \ldots, s_n^* \rangle$ is a **Nash equilibrium** if for any agent $i$, $s_i^*$ is the best response to $\langle s_1^*, s_2^*, \ldots, s_{i-1}^*, s_{i+1}^*, \ldots, s_n^* \rangle$.

What are the Nash equilibria in the Battle of sexes? What about the matching pennies?
1. Required Concepts

2. Non-Coop Games in NF

**Theorem 1.20 (Nash (1950))**

Every finite normal form game has a Nash equilibrium.

**Corollary 1.21 (Nash Eq. in constant-sum Games)**

In any finite normal form 2-person constant-sum game, the Nash equilibria are exactly all pairs \( \langle s_1, s_2 \rangle \) of maxmin strategies (\( s_1 \) for player 1, \( s_2 \) for player 2).

All Nash equilibria have the same payoff (expected utility): the value of the game, that player 1 gets.
1.3 Non-Coop Extensive Games
We have previously introduced normal form games (Definition 1.5). This notion does not allow to deal with sequences of actions that are reactions to actions of the opponent.

**Extensive form (tree form) games**

Unlike games in normal form, those in extensive form do not assume that all moves between players are made simultaneously. This leads to a tree form, and allows to introduce strategies, that take into account the history of the game.

We distinguish between perfect and imperfect information games. While the former assume that the players have complete knowledge about the game, the latter do not: a player might not know exactly which node it is in.
The following definition covers a game as a tree:

**Definition 1.22 (Extensive form Games, Perfect Inf.)**

A finite perfect information game in extensive form is a tuple \( G = \langle \mathcal{A}, A, H, Z, \alpha, \rho, \sigma, \mu_1, \ldots, \mu_n \rangle \) where

- \( \mathcal{A} \) is a set of \( n \) players, \( A \) is a set of actions
- \( H \) is a set of non-terminal nodes, \( Z \) a set of terminal nodes, \( H \cap Z = \emptyset \),
- \( \alpha : H \to A \) assigns to each node a set of actions,
- \( \rho : H \to N \) assigns to each non-terminal node a player who chooses an action at that node,
- \( \sigma : H \times A \to H \cup Z \) assigns to each (node, action) a successor node \( (h_1 \neq h_2 \implies \sigma(h_1, a_1) \neq \sigma(h_2, a_2)) \),
- \( \mu_i : Z \to \mathbb{R} \) are the utility functions.
Such games can be visualised as finite trees. Here is the famous “Sharing Game”.

![Figure 2: The Sharing game.](image)
Here is another (generic) game.

Figure 3: A generic game.
Transforming extensive form games into normal form

Note that the definitions of best response and Nash equilibria carry over to games in extensive form. But they do not take into account the sequential nature of extensive games. Indeed we have:

**Lemma 1.23 (Extensive form $\leftrightarrow$ Normal form)**

*Each game in extensive form can be transformed in a normal form game (such that the strategy spaces are the same).*

The idea is to take the *set of all strategies of agent* $i$ as the *set of actions of agent* $i$. And to define the utility function accordingly.
Transforming the generic game into normal form

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE</td>
<td>W</td>
<td>Y</td>
</tr>
<tr>
<td>AF</td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>BE</td>
<td>Z</td>
<td>Z</td>
</tr>
<tr>
<td>BF</td>
<td>Z</td>
<td>Z</td>
</tr>
</tbody>
</table>
Is there a converse of Lemma 1.23?

We consider prisoner’s dilemma and try to model a game in extensive form with the same payoffs and strategy profiles.
In fact, it is not surprising that we do not succeed in the general case:

**Theorem 1.24 (Zermelo, 1913)**

*Every perfect information game in extensive form has a pure strategy Nash equilibrium.*

We will later introduce imperfect information games (in extensive form): Slide 53.
Example 1.25 (Unintended Nash equilibria)

We consider the following perfect information game in extensive form.

Figure 4: Unintended Equilibrium.
Both \((A, R)\) as well as \((B, L)\) are Nash equilibria.

But \((B, L)\) is **unintuitive**.

This leads to the notion of **subgame perfect Nash equilibria**:

**Definition 1.26 (Subgame perfect Nash equilibria)**

Let \(G\) be a perfect information game in extensive form.

*Subgame:* A subgame of \(G\) rooted at node \(h\) is the restriction of \(G\) to the descendants of \(h\).

*SPE:* The **subgame perfect Nash equilibria** of a perfect information game \(G\) in extensive form are those Nash equilibria of \(G\), that are also Nash equilibria for all subgames \(G'\) of \(G\).
1. Required Concepts

3. Non-Coop Extensive Games

Figure 5: The Centipede game.

- The only SPE is for each player to go down.
- But many human players would rather go across and hope for a better payoff.
**Definition 1.27 (Extensive form Games, Imperfect Inf.)**

A finite imperfect information game in extensive form is a tuple

\[ G = \langle A, A, H, Z, \alpha, \rho, \sigma, \mu_1, \ldots, \mu_n, I_1, \ldots, I_n \rangle \]

where

- \( \langle A, A, H, Z, \alpha, \rho, \sigma, \mu_1, \ldots, \mu_n \rangle \) is a perfect information game in the sense of Definition 1.22 on Slide 43,

- \( I_i \) are equivalence relations on \( \{ h \in H : \rho(h) = i \} \) such that \( h, h' \in I_i \) implies \( \alpha(h) = \alpha(h') \).
Figure 6: An imperfect game.
Definition 1.28 (Pure strategy in imperfect info. games)

Given an imperfect information game in extensive form, a pure strategy for player \( i \) is a vector \( \langle a_1, \ldots, a_k, \rangle \) with \( a_j \in \alpha(I_{(i,j)}) \) where \( I_{(i,1)}, \ldots I_{(i,k)} \) are the \( k \) equivalence classes for agent \( i \).
There is a pure strategy Nash equilibrium.

Figure 7: Prisoner’s dilemma.
NF game $\leftrightarrow$ Imperfect game

Each game in normal form can be transformed into an imperfect information game in extensive form.

Each imperfect information game in extensive form can be transformed into a game in normal form.

This is obvious if we consider pure strategies. But what about mixed strategies?

What is the set of mixed strategies for an imperfect game?
SPE: What about subperfect equilibria (analogue of Definition 1.26 on Slide 51 for imperfect games?)

First try: In each information set, we have a set of subgames (a forest). Why not asking that a strategy should be a best response in all subgames of that forest?
Figure 8: Subgames in Imperfect Games.
Nash equilibria: \((L, U)\) and \((R, D)\).

In one subgame, \(U\) dominates \(D\), in the other \(D\) dominates \(U\).

But \((R, D)\) seems to be the unique choice: both players can put themselves into the others place and reason accordingly.

Requiring that a strategy is best response to all subgames is too strong.
1. Required Concepts

*Game Theory*. MIT Press.

Equilibrium points in n-person games. 
*Proceedings of the National Academy of Sciences of the United States of America* 36, 48–49.

*A Course in Game Theory*. MIT Press.

A class of games possessing pure-strategy Nash equilibria. 

Chapter 2. How to Form a Coalition

How to Form a Coalition

2.1 Coalitional Games
2.2 Coalition-Structure-Search
2.3 Core versus Shapley Value
2.4 Computational Issues
2.5 References
2. How to Form a Coalition

Outline

We consider in this chapter *coalitional games* (also called *cooperative*). In contrast to the previous chapter, the basic notion is a *team of agents* that work together, not a single agent, as in non-cooperative games.

- discuss two solution concepts: the *core* and the *Shapley value*.
- describe an important anytime algorithm: CSS, the *coalition structure search*,
- discuss the *relations* between the core and the Shapley value, and
- state some *complexity results* (computing the Shapley value, is the core empty?).
2. How to Form a Coalition

Idea: Consider a protocol (to build coalitions) as a game and consider Nash-equilibria.

Problem: Nash-Eq is too weak!

Definition 2.1 (Strong Nash Equilibrium)

A profile is in **strong Nash-Eq** if there is no subgroup that can deviate by changing strategies jointly in a manner that increases the payoff of all its members, given that nonmembers stick to their original choice.

This is often too strong and does not exist.
2. How to Form a Coalition

1. Coalitional Games

2.1 Coalitional Games
Definition 2.2 (Coalitional Game (CFG))

A **coalitional game**, also called characteristic function game (CFG) is a pair $\langle A, v \rangle$ where $v : 2^A \rightarrow \mathbb{R}$ is a function assigning each coalition a real-valued number.

Thus it is independent of the nonmembers.
Is that really true?

1. **Positive Externalities**: Overlapping goals. Nonmembers perform actions and move the world closer to the coalition’s goal state.

2. **Negative Externalities**: Shared resources. Nonmembers may use the resources so that not enough is left.
Definition 2.3 (Coalition Formation in CFG’s)

Coalition Formation in CFG’s consists of:

**Forming CS:** Formation of coalitions such that within each coalition agents coordinate their activities. This partitioning is called *coalition structure CS*.

**Solving Optimisation Problem:** For each coalition the tasks and resources of the agents have to be pooled. *Maximise monetary value.*

**Payoff Division:** Divide the value of the generated solution among agents.
Definition 2.4 (Additive, Simple Games)

A game \( \langle A, v \rangle \) is called **additive**, if

\[ v_{S \cup T} = v_S + v_T, \]

for all \( S, T \subseteq A \) with \( S \cap T = \emptyset \).

A game is called **simple**, if for every \( S \subseteq A \) the following holds: \( v_S \in \{0, 1\} \).

Lemma 2.5

Additive games are constant sum games.
2. How to Form a Coalition

1. Coalitional Games

Definition 2.6 (Super-additive Games)

A game \( \langle A, \nu \rangle \) is called super-additive, if

\[
\nu_{S \cup T} \geq \nu_S + \nu_T,
\]

where \( S, T \subseteq A \) and \( S \cap T = \emptyset \).

Lemma 2.7

Coalition formation for super-additive games is trivial.

Conjecture

All games are super-additive.
The conjecture is wrong, because the coalition process is not for free: **communication costs, penalties, time limits.**

**Definition 2.8 (Sub-additive Games)**

A game $\langle A, v \rangle$ is called **sub-additive**, if

$$v_{S \cup T} \leq v_S + v_T,$$

where $S, T \subseteq A$ and $S \cap T = \emptyset$.

Coalition formation for sub-additive games is trivial.
2. How to Form a Coalition

1. Coalitional Games

**Definition 2.9 (Convex Game)**

A game \( \langle A, v \rangle \) is called **convex**, if the following holds

\[
v_S + v_T \leq v_{S \cup T} + v_{S \cap T}
\]

for all \( S, T \subseteq A \).

*Obviously, this implies super-additivity.*

**Definition 2.10 (Veto Player)**

A player \( i \) is called **veto player**, if \( v_{A \setminus \{i\}} = 0 \).
Example 2.11 (Treasure of Sierra Madre Game)
There are \( n \) people finding a treasure of many gold pieces in the Sierra Madre. Each piece can be carried by two people, not by a single person.

Example 2.12 (3-player Majority Game)
There are 3 people that need to agree on something. If they all agree, there is a payoff of 1. If just 2 agree, they get a payoff of \( \alpha \) (\( 0 \leq \alpha \leq 1 \)). The third player gets nothing.

Example 2.13 (Parliament Game)
The parliament has to decide about passing a 100 million Euro spending bill. There are 4 parties with the following number of representatives: A: 45, B: 25, C:15, and D:15. The bill passes when at least 51 vote for it.
How do the $v_S$ look like?

In the Sierra Madre Game: $v_S = \left\lfloor \frac{|S|}{2} \right\rfloor$.

For the Majority Game:

$$v_S = \begin{cases} 
1, & \text{if } S = \{1, 2, 3\}; \\
\alpha, & \text{if } |S| = 2; \\
0, & \text{if } |S| = 1.
\end{cases}$$

For the Parliament Game:

$$100 = v\{A,B,C,D\} = v\{A,B\} = v\{A,C\} = v\{A,D\} = v\{B,C,D\}$$

and

$$0 = v\{B,C\} = v\{B,D\} = v\{C,D\}.$$
2.2 Coalition-Structure-Search
Maximise the social welfare of the agents $\mathcal{A}$ by finding a coalition structure

$$\mathcal{C}S^* = \arg \max_{\mathcal{C}S \in \text{part}(\mathcal{A})} \text{Val}(\mathcal{C}S),$$

where

$$\text{Val}(\mathcal{C}S) := \sum_{S \in \mathcal{C}S} v_S.$$

How many coalition structures are there?
Let $Z(|A|, i)$ denote the number of coalition structures with $i$ coalitions. Then

- $Z(|A|, |A|) = Z(|A|, 1) = 1$.
- $Z(|A|, i) = iZ(|A| - 1, i) + Z(|A| - 1, i - 1)$.

Add one agent to a game with $|A| - 1$ agents.

$\sum_{i=1}^{|A|} Z(|A|, i)$ is the number of coalition structures.

This is in the order of $|A|^2$. 

Too many: $\Omega(|A|^{|A|/2})$. Enumerating is only feasible if $|A| < 15$.

Figure 9: Number of Coalition (Structures).
How can we approximate $\text{Val}(CS)$?

Choose set $\mathcal{N}$ (a subset of all partitions of $A$) and pick the best coalition seen so far:

$$
CS^*_\mathcal{N} = \arg \max_{CS \in \mathcal{N}} \text{Val}(CS).
$$
Figure 10: Coalition Structure Graph.
We want our approximation as good as possible.

That means:

\[
\frac{\text{Val}(\mathcal{CS}^*)}{\text{Val}(\mathcal{CS}_{\mathcal{N}}^*)} \leq k,
\]

where \( k \) is as small as possible.
2. How to Form a Coalition

We consider 3 search algorithms:

**MERGE:** Breadth-first search from the top.

**SPLIT:** Breadth first from the bottom.

**Coalition-Structure-Search (CSS1):** First the bottom 2 levels are searched, then a breadth-first search from the top.

MERGE might **not even get a bound**, without looking at all coalitions.

SPLIT gets a good bound \((k = |A|)\) after searching the bottom 2 levels (see below). But then it **can get slow**.

CSS1 **combines** the good features of MERGE and SPLIT.
Why is SPLIT slow after the first two bottom levels? Construct a bad example as follows.

\[
\nu_S = \begin{cases} 
1, & \text{if } |S| = 1; \\
0, & \text{otherwise.} 
\end{cases}
\]

So the optimum is the top node, and

\[
\frac{\text{Val}(CS^*)}{\text{Val}(CS_N^*)} = \frac{|A|}{l - 1},
\]

where \(l\) is the level that the algorithm has completed (the number of unit coalitions on a level \(l\) is always \(\leq l - 1\) (except the top level where it is equal to \(l\), namely \(|A|\)).
Theorem 2.14 (Minimal Search to get a bound)

To bound $k$, it suffices to search the lowest two levels of the CS-graph. Using this search, the bound $k = |\mathcal{A}|$ can be taken. This bound is tight and the number of nodes searched is $2^{|\mathcal{A}|-1}$.

No other search algorithm can establish the bound $k$ while searching through less than $2^{|\mathcal{A}|-1}$ nodes.
Proof.

There are at most $|A|$ coalitions included in $CS^*$. Thus

$$\text{Val}(CS^*) \leq |A| \max_S v_S \leq |A| \max_{CS \in \mathcal{N}} \text{Val}(CS) = |A| \text{Val}(CS^*_\mathcal{N})$$

Number of coalitions at the second lowest level: $2|A| - 2$.

Number of coalition structures at the second lowest level: $\frac{1}{2}(2|A| - 2) = 2|A|^{-1} - 1$.

Thus the number of nodes visited is: $2|A|^{-1}$.  \qed
What exactly does the last theorem mean? Let \( n_{\text{min}} \) be the smallest size of \( \mathcal{N} \) such that a bound \( k \) can be established.

**Positive result:** \( \frac{n_{\text{min}}}{\text{partitions of } \mathcal{A}} \) approaches 0 for \( |\mathcal{A}| \to \infty \).

**Negative result:** To determine a bound \( k \), one needs to search through exponentially many coalition structures.
2. How to Form a Coalition

2. Coalition-Structure-Search

Algorithm (\texttt{CS-Search-1})

The algorithm comes in 3 steps:

1. Search the bottom two levels of the \texttt{CS}-graph.

2. Do a breadth-first search from the top of the graph.

3. Return the \texttt{CS} with the highest value.

This is an \textit{anytime algorithm}. 
Theorem 2.15 (CS-Search-1 up to Layer l)

With the algorithm CS-Search-1 we get the following bound for $k$ after searching through layer $l$:

$$
\begin{cases}
\left\lceil \frac{|A|}{h} \right\rceil & \text{if } |A| \equiv h - 1 \mod h \text{ and } |A| \equiv l \mod 2, \\
\left\lfloor \frac{|A|}{h} \right\rfloor & \text{otherwise}.
\end{cases}
$$

where $h = \text{def } \left\lfloor \frac{|A| - l}{2} \right\rfloor + 2$.

Thus, for $l = |A|$ (check the top node), $k$ switches from $|A|$ to $\frac{|A|}{2}$. 
1. Is **CS-Search-1** the **best anytime algorithm**?

2. The search for best $k$ for $n' > n$ is perhaps not the same search to get best $k$ for $n$.

3. **CS-Search-1** does not use any information while searching. Perhaps $k$ can be made smaller by not only considering $\text{Val}(CS)$ but also $v_S$ in the searched $CS'$. 
2.3 Core versus Shapley Value
From now on we assume super-additivity!

**Definition 2.16 (Payoff Vector, Core of a game)**

A **payoff vector** for a CFG is a tuple \( \langle x_1, \ldots, x_n \rangle \) such that \( x_i \geq 0 \) and \( \sum_{i=1}^{n} x_i = v_A \).

The **core of a CFG** is the set of all payoff vectors such that the following holds:

\[
\forall S \subseteq A : \sum_{i \in S} x_i \geq v_S
\]

(core corresponds to strong Nash equilibrium)

What about the core in the three examples from Slide 74? Core
Sierra Madre:

- **Case 1:** $|A| \geq 4$ and $|A|$ is even. Then the core consists of a single payoff vector $\langle \frac{1}{2}, \ldots, \frac{1}{2} \rangle$.

- **Case 2:** $|A| \geq 3$ and $|A|$ is odd. Then the core is empty.
3 Player Majority Game: The core consists of all payoff vectors that assign 1 to the grand coalition and something greater than $\alpha$ to all coalitions with two agents. Thus we have three cases:

- **Case 1:** $\alpha \leq \frac{2}{3}$. Then the core consists of the payoff vectors $\langle \frac{\alpha}{2}, \frac{\alpha}{2}, 1 - \alpha \rangle$, $\langle \frac{\alpha}{2}, 1 - \alpha, \frac{\alpha}{2} \rangle$, $\langle 1 - \alpha, \frac{\alpha}{2}, \frac{\alpha}{2} \rangle$.

- **Case 2:** $\alpha = \frac{2}{3}$. Then the core consists of the vector $\langle \frac{1}{3}, \frac{1}{3}, \frac{1}{3} \rangle$.

- **Case 3:** $\alpha \geq \frac{2}{3}$. Then the core is empty.
Parliament Game: Consider payoff vectors $\langle a, b, c, d \rangle$. Whenever one of $a, b, c, d$ is non-zero, then we can build the coalition consisting of the remaining three parties which gets a higher payoff. Therefore the core is empty.
2. How to Form a Coalition

3. Core versus Shapley Value

**Definition 2.17 (Shapley-Value)**

The **Shapley value** of agent $i$ is defined by

$$x_i = \frac{1}{|\mathcal{A}|!} \sum_{S \subseteq \mathcal{A}} (|\mathcal{A}| - |S| - 1)!|S|!(v_{S \cup \{i\}} - v_S)$$

View coalition formation by **adding one agent at a time**.

For a given sequence (of how to add agents), what is agent $i$’s **marginal contribution** (at the time it is added to the set $S$)?

It is $(v_{S \cup \{i\}} - v_S)$.

- How many ways to form $S$ (before $i$ joined): $|S|!$.
- How many ways to form $S'$ (after $i$ joined): $(|\mathcal{A}| - |S| - 1)!$.
- This has to be summed up over all possible sets $S$ and averaged over all $|\mathcal{A}|!$ orderings of the agents.
What are the Shapley values for the Parliament Game?

- $B, C, D$ should all get the same amount (why?).
- $A$ should get more.
- Doing the math gives: $A$: 50, and the remaining 3 each get $16\frac{2}{3}$. 
Theorem 2.18 ((Non-) Emptiness of the Core)

1. The core of a constant-sum game that is not additive, is empty.

2. In a simple game, the core is empty if and only if there is no veto player. When there are veto players, the core consists of all payoff vectors in which the non-veto players get 0.

3. Convex games have non-empty cores.

4. For convex games, the vector of Shapley values belongs to the core.
The payoff division should be fair between the agents, otherwise they leave the coalition.

**Definition 2.19 (Dummies, Interchangeable)**

Agent \( i \) is called a **dummy**, if

\[
\text{for all coalitions } S \text{ with } i \not\in S: v_{S \cup \{i\}} - v_S = v_i.
\]

Agents \( i \) and \( j \) are called **interchangeable**, if

\[
\text{for all coalitions } S \text{ with } i \in S \text{ and } j \not\in S: v_{S \setminus \{i\} \cup \{j\}} = v_S
\]
Three axioms:

**Symmetry:** If $i$ and $j$ are interchangeable, then

$$x_i = x_j.$$

**Dummies:** For all dummies $i$: $x_i = v_i$.

**Additivity:** For any two games $v, w$:

$$x_{i}^{v \oplus w} = x_{i}^{v} + x_{i}^{w},$$

where $v \oplus w$ denotes the game defined by $(v \oplus w)_S = v_S + w_S$. 
Theorem 2.20 (Shapley-Value)

There is only one payoff division satisfying the above three axioms: The Shapley value from Definition 2.17.
2.4 Computational Issues
A coalitional game specifies for each coalition a value. So the representation is already exponential in the number of agents.

Using the Stirling formula, it can be shown that the Shapley value can be computed in time $O(N^{\log \log n})$, where $N = 2^n$ is the input size.

We need to find a more succinct game representation.
Definition 2.21 (Weighted Graph Game (WGG))

Let $\langle V, W \rangle$ be an undirected weighted graph ($V$ the set of vertices, $W \in \mathbb{R}^{V \times V}$ the set of edge weights).

The associated weighted graph game $\langle A, v \rangle$ is defined as follows:

- $A = V,$
- $v_S := \sum_{i,j \in S} w(i, j).$

Real life example?
Lemma 2.22

Let a weighted graph game be given, where all weights are non-negative. Then this game is **convex** and membership of a payoff vector in the core can be tested in polynomial time.
Theorem 2.23 (Shapley Value of a WGG)

The Shapley value of a weighted graph game \( \langle V, W \rangle \) is given by

\[
x_i = \frac{1}{2} \sum_{i \neq j} w(i, j)
\]

Therefore the **Shapley value can be computed in quadratic time.**
Theorem 2.24 (Nonemptiness of the core of a WGG)

The problem to decide whether the core is empty for a Weighted Graph Game, is NP-complete.
2. How to Form a Coalition

5. References


Chapter 3. Reasoning about Coalitions

3.1 Modal Logic
3.2 ATL
3.3 Rational Play (ATLP)
3.4 Imperfect Information
3.5 Model Checking
3.6 References
3. Reasoning about Coalitions

Outline

- In the previous chapter, we showed how coalitions can be rationally formed,
- In this chapter, we show how one can use modal logic to reason about their play and their outcome.
3. Reasoning about Coalitions

1. Modal Logic

3.1 Modal Logic
Why logic at all?

- framework for **thinking** about systems,
- makes one **realise** the implicit **assumptions**,
- and then we can:
  - **investigate** them, **accept or reject** them,
  - **relax** some of them and still use a part of the formal and conceptual machinery;
- reasonably expressive but simpler and more rigorous than the full language of mathematics.
Why logic at all?

- **verification**: check *specification* against implementation,
- **executable specification**, 
- **planning as model checking**
Modal logic is an extension of classical logic by new connectives □ and ◊: necessity and possibility.

- “□p is true” means \( p \) is necessarily true, i.e. true in every possible scenario,
- “◊p is true” means \( p \) is possibly true, i.e. true in at least one possible scenario.
Various modal logics:

- knowledge $\rightarrow$ **epistemic logic**,
- beliefs $\rightarrow$ **doxastic logic**,
- obligations $\rightarrow$ **deontic logic**,
- actions $\rightarrow$ **dynamic logic**,
- time $\rightarrow$ **temporal logic**,
- and **combinations of the above**: most famous **multimodal logics**:
  - **BDI logics** of beliefs, desires, intentions (and time).
3. Reasoning about Coalitions

1. Modal Logic

**Definition 3.1 (Kripke Semantics)**

Kripke model (possible world model):

\[ M = \langle \mathcal{W}, R, \pi \rangle, \]

- \( \mathcal{W} \) is a set of possible worlds
- \( R \subseteq \mathcal{W} \times \mathcal{W} \) is an accessibility relation
- \( \pi : \mathcal{W} \rightarrow \mathcal{P}(\Pi) \) is a valuation of propositions.

\[ M, w \models \Box \varphi \quad \text{iff for every } w' \in \mathcal{W} \text{ with } wRw' \text{ we have that } M, w' \models \varphi. \]
3. Reasoning about Coalitions

1. Modal Logic

An Example

\[ x = 1 \rightarrow K_s x = 1 \]
3.2 ATL
ATL: What Agents Can Achieve

- **ATL: Agent Temporal Logic** [Alur et al. 1997]
- Temporal logic meets game theory
- Main idea: cooperation modalities

\[ \langle A \rangle \Phi: \text{coalition } A \text{ has a collective strategy to enforce } \Phi \]
3. Reasoning about Coalitions

2. ATL

- $\langle \text{jamesbond} \rangle \Diamond \text{win}$: “James Bond has an infallible plan to eventually win”

- $\langle \text{jamesbond}, \text{bondsgirl} \rangle \text{fun}\mathcal{U} \text{shot}$: “James Bond and his girlfriend are able to have fun until someone shoots at them”

- “Vanilla” ATL: every temporal operator preceded by exactly one cooperation modality;

- ATL*: no syntactic restrictions;
ATL Models: Concurrent Game Structures

- Agents, actions, transitions, atomic propositions
- Atomic propositions + interpretation
- Actions are abstract
Definition 3.2 (Concurrent Game Structure)

A **concurrent game structure** is a tuple \( M = \langle \text{Ag}t, Q, \pi, \text{Act}, d, o \rangle \), where:

- \( \text{Ag}t \): a finite set of all **agents**
- \( Q \): a set of **states**
- \( \pi \): a **valuation** of propositions
- \( \text{Act} \): a finite set of (atomic) **actions**
- \( d : \text{Ag}t \times Q \rightarrow \mathcal{P}(\text{Act}) \) defines actions available to an agent in a state
- \( o \): a deterministic **transition function** that assigns outcome states \( q' = o(q, \alpha_1, \ldots, \alpha_k) \) to states and tuples of actions
Example: Robots and Carriage
Definition 3.3 (Strategy)

A strategy is a conditional plan. We represent strategies by functions \( s_a : Q \rightarrow \text{Act} \).

Function \( \text{out}(q, S_A) \) returns the set of all paths that may result from agents \( A \) executing strategy \( S_A \) from state \( q \) onward.
### Definition 3.4 (Semantics of ATL)

<table>
<thead>
<tr>
<th>Expression</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M, q \models p$</td>
<td>iff $p$ is in $\pi(q)$;</td>
</tr>
<tr>
<td>$M, q \models \varphi \land \psi$</td>
<td>iff $M, q \models \varphi$ and $M, q \models \psi$;</td>
</tr>
<tr>
<td>$M, q \models \langle A \rangle \Phi$</td>
<td>iff <strong>there is a collective strategy</strong> $S_A$ such that, for every path $\lambda \in \text{out}(q, S_A)$, we have $M, \lambda \models \Phi$.</td>
</tr>
<tr>
<td>$M, \lambda \models \bigcirc \varphi$</td>
<td>iff $M, \lambda[1] \models \varphi$;</td>
</tr>
<tr>
<td>$M, \lambda \models \Diamond \varphi$</td>
<td>iff $M, \lambda[i] \models \varphi$ for some $i \geq 0$;</td>
</tr>
<tr>
<td>$M, \lambda \models \Box \varphi$</td>
<td>iff $M, \lambda[i] \models \varphi$ for all $i \geq 0$;</td>
</tr>
<tr>
<td>$M, \lambda \models \varphi U \psi$</td>
<td>iff $M, \lambda[i] \models \psi$ for some $i \geq 0$, and $M, \lambda[j] \models \varphi$ for all $0 \leq j \leq i$.</td>
</tr>
</tbody>
</table>
Example: Robots and Carriage

\[
\text{pos}_0 \rightarrow \langle \langle 1 \rangle \rangle \square \neg \text{pos}_1
\]
Temporal operators allow a number of useful concepts to be formally specified

- safety properties
- liveness properties
- fairness properties
Safety (maintenance goals):

“something bad will not happen”
“something good will always hold”

Typical example:

□¬bankrupt

Usually: □¬....

In ATL:

⟨⟨os⟩⟩□¬crash
Liveness (achievement goals): “something good will happen”

Typical example:

◇ rich

Usually: ◇ ....

In ATL:

\(\langle \langle alice, bob \rangle \rangle \Diamond \text{paperAccepted} \)
Fairness (service goals):

“if something is attempted/requested, then it will be successful/allocated”

Typical examples:

□(attempt  →  ◊success)
□◊attempt  →  □◊success

In ATL* (!):

⟨⟨prod, dlr⟩⟩□(carRequested  →  ◊carDelivered)
Connection to Games

- Concurrent game structure = generalized extensive game

- $\langle \langle A \rangle \rangle \gamma$: $\langle \langle A \rangle \rangle$ splits the agents into proponents and opponents

- $\gamma$ defines the winning condition

- $\leadsto$ infinite 2-player, binary, zero-sum game

- Flexible and compact specification of winning conditions
Solving a game \( \approx \) checking if \( M, q \models \langle \langle A \rangle \rangle \gamma \)

But: do we really want to consider all the possible plays?
3.3 Rational Play (ATLP)
Game-theoretical analysis of games:

- *Solution concepts* define rationality of players
  - maxmin
  - Nash equilibrium
  - subgame-perfect Nash
  - undominated strategies
  - Pareto optimality

- Then: we assume that players play rationally
- ...and we ask about the outcome of the game under this assumption

- Role of rationality criteria: *constrain the possible game moves* to “sensible” ones
3. Reasoning about Coalitions

3. Rational Play (ATLP)

\[ \text{start} \rightarrow \neg \langle 1 \rangle \diamond \text{money}_1 \]

\[ \text{start} \rightarrow \neg \langle 2 \rangle \diamond \text{money}_2 \]
ATL + Plausibility (ATLP)

**ATL:** reasoning about *all* possible behaviors.

$\langle \langle A \rangle \rangle \varphi$: agents $A$ have *some* collective strategy to enforce $\varphi$ against any response of their opponents.

**ATLP:** reasoning about *plausible* behaviors.

$\langle \langle A \rangle \rangle \varphi$: agents $A$ have a *plausible* collective strategy to enforce $\varphi$ against any plausible response of their opponents.

**Important**

The possible strategies of both $A$ and $\text{Agt} \setminus A$ are restricted.
New in ATLP:

\((\text{set-pl } \omega) : \text{the set of plausible profiles is set/reset to the strategies described by } \omega. \)  
Only plausible strategy profiles are considered!

Example: \((\text{set-pl } greedy_1)\langle 2 \rangle \diamond money_2\)
Concurrent game structures with plausibility

\[ M = (\text{Agt}, Q, \Pi, \pi, Act, d, \delta, \Upsilon, \Omega, \| \cdot \|) \]

- \( \Upsilon \subseteq \Sigma \): set of (plausible) strategy profiles

- \( \Omega = \{ \omega_1, \omega_2, \ldots \} \): set of plausibility terms
  
  **Example:** \( \omega_{NE} \) may stand for all Nash equilibria

- \( \| \cdot \| : Q \rightarrow (\Omega \rightarrow \mathcal{P}(\Sigma)) \): plausibility mapping
  
  **Example:** \( \| \omega_{NE} \|_q = \{ (\text{confess}, \text{confess}) \} \)
3. Reasoning about Coalitions

Outcome = Paths that may occur when agents $A$ perform $s_A$ when only plausible strategy profiles from $\Upsilon$ are played

\[
out_{\Upsilon}(q, s_A) = \{ \lambda \in Q^+ \mid \exists t \in \Upsilon(s_A) \forall i \in \mathbb{N} (\lambda[i + 1] = \delta(\lambda[i], t(\lambda[i]))) \}
\]

$P$: the players always show same sides of their coins

$s_1$: always show “heads”
Semantics of ATLP

\[ M, q \models \langle A \rangle\gamma \text{ iff there is a strategy } s_A \text{ consistent with } \gamma \text{ such that } M, \lambda \models \gamma \text{ for all } \lambda \in \text{out}_\gamma(q, s_A) \]

\[ M, q \models (\text{set-pl } \omega)\varphi \text{ iff } M^\omega, q \models \varphi \text{ where the new model } M^\omega \text{ is equal to } M \text{ but the new set } \gamma^\omega \text{ of plausible strategy profiles is set to } \| \omega \|_q. \]
Example: Pennies Game

What is a Nash equilibrium in this game?

We need some kind of winning criteria!
3. Reasoning about Coalitions

Agent 1 “wins”, if $\gamma_1 \equiv \Box(\neg \text{start} \rightarrow \text{money}_1)$ is satisfied.
Agent 2 “wins”, if $\gamma_2 \equiv \Diamond \text{money}_2$ is satisfied.

Now we have a qualitative notion of success.

$M, q_0 \models (\text{set-pl } \omega_{NE}) \langle 2 \rangle \Box(\neg \text{start} \rightarrow \text{money}_1)$

where $\parallel \omega_{NE} \parallel_{q_0} = “\text{all profiles belonging to grey cells”}$.
How to obtain plausibility terms?

Idea

Formulae that describe plausible strategies!

\[(\text{set-pl } \sigma.\theta)\varphi: \text{ “suppose that } \theta \text{ characterizes rational strategy profiles, then } \varphi \text{ holds”}].\]

Sometimes quantifiers are needed...

E.g.: \[(\text{set-pl } \sigma. \forall \sigma' \text{ dominates}(\sigma, \sigma'))\]
Characterization of Nash Equilibrium

$\sigma_a$ is $a$’s best response to $\sigma$ (wrt $\vec{\gamma}$):

$$BR^\vec{\gamma}_a(\sigma) \equiv (\text{set-pl } \sigma[\text{Ag}t\setminus\{a\}])(\langle a \rangle \gamma_a \rightarrow (\text{set-pl } \sigma)\langle \emptyset \rangle \gamma_a)$$

$\sigma$ is a Nash equilibrium:

$$NE^\vec{\gamma}(\sigma) \equiv \bigwedge_{a \in \text{Ag}t} BR^\vec{\gamma}_a(\sigma)$$
Example: Pennies Game revisited

\( \gamma_1 \equiv \Box(\neg \text{start} \rightarrow \text{money}_1) \); \( \gamma_2 \equiv \Diamond \text{money}_2 \).

\[
M_1, q_0 \models (\text{set-pl } \sigma.NE^{\gamma_1, \gamma_2}(\sigma))\langle 2 \rangle \Box(\neg \text{start} \rightarrow \text{money}_1)
\]

...where \( NE^{\gamma_1, \gamma_2}(\sigma) \) is defined as on the last slide.
Characterizations of Other Solution Concepts

σ is a subgame perfect Nash equilibrium:

\[ SPN^{\gamma}(\sigma) \equiv \langle \emptyset \rangle \Box NE^{\gamma}(\sigma) \]

σ is Pareto optimal:

\[ PO^{\gamma}(\sigma) \equiv \forall \sigma' \left( \right. \right.

\[ \bigwedge_{a \in \text{Agt}} ((\text{set-pl } \sigma') \langle \emptyset \rangle \gamma_a \rightarrow (\text{set-pl } \sigma) \langle \emptyset \rangle \gamma_a) \lor \]

\[ \bigvee_{a \in \text{Agt}} ((\text{set-pl } \sigma) \langle \emptyset \rangle \gamma_a \land \neg (\text{set-pl } \sigma') \langle \emptyset \rangle \gamma_a) \bigg). \]
σ is undominated:

\[ \text{UNDOM}^\gamma(\sigma) \equiv \forall \sigma_1 \forall \sigma_2 \exists \sigma_3 \]

\[
\left( \left( \text{set-pl} \left\langle \sigma_1 \{a\}, \sigma_2^{\text{Agt}\setminus\{a\}} \right\rangle \langle \emptyset \rangle \gamma_a \rightarrow \right.
\]

\[
\left. \text{set-pl} \left\langle \sigma_1 \{a\}, \sigma_2^{\text{Agt}\setminus\{a\}} \right\rangle \langle \emptyset \rangle \gamma_a \right)
\]

\[
\lor \left( \text{set-pl} \left\langle \sigma_1 \{a\}, \sigma_3^{\text{Agt}\setminus\{a\}} \right\rangle \langle \emptyset \rangle \gamma_a \land \right.
\]

\[
\left. \neg \text{set-pl} \left\langle \sigma_1 \{a\}, \sigma_3^{\text{Agt}\setminus\{a\}} \right\rangle \langle \emptyset \rangle \gamma_a \right) \right).
Theorem 3.5

The characterizations coincide with game-theoretical solution concepts in the class of game trees.
3.4 Imperfect Information
How can we reason about extensive games with imperfect information?

Let’s put ATL and epistemic logic in one box.

⇝ Problems!
3. Reasoning about Coalitions

4. Imperfect Information

\[(\sim, \sim) \rightarrow q_0 \rightarrow q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_4 \rightarrow q_5 \rightarrow q_6 \rightarrow q_7 \rightarrow q_8 \rightarrow q_9 \rightarrow q_{10} \rightarrow q_{11} \rightarrow q_{12} \rightarrow q_{13} \rightarrow q_{14} \rightarrow q_{15} \rightarrow q_{16} \rightarrow q_{17} \rightarrow q_{18} \rightarrow \text{win} \]

\[
\begin{align*}
\text{start} & \rightarrow \langle \langle a \rangle \rangle \Diamond \text{win} \\
\text{start} & \rightarrow K_a \langle \langle a \rangle \rangle \Diamond \text{win} \\
\end{align*}
\]

Does it make sense?
3. Reasoning about Coalitions

Problem:
Strategic and epistemic abilities are not independent!

\[ \langle A \rangle \Phi = A \text{ can enforce } \Phi \]

It should at least mean that \( A \) are able to identify and execute the right strategy!

Executable strategies = uniform strategies
Definition 3.6 (Uniform strategy)

Strategy $s_a$ is **uniform** iff it specifies the same choices for indistinguishable situations:

- (no recall:) if $q \sim_a q'$ then $s_a(q) = s_a(q')$
- (perfect recall:) if $\lambda \approx_a \lambda'$ then

  $\Rightarrow s_a(\lambda) = s_a(\lambda)$, where $\lambda \approx_a \lambda'$ iff $\lambda[i] \sim_a \lambda'[i]$ for every $i$.

A collective strategy is uniform iff it consists only of uniform individual strategies.
Note:

Having a successful strategy does not imply knowing that we have it!

Knowing that a successful strategy exists does not imply knowing the strategy itself!
Levels of Strategic Ability

From now on, we restrict our discussion to uniform memoryless strategies.

Our cases for $\langle A \rangle \Phi$ under incomplete information:

2. There is $\sigma$ such that, for every execution of $\sigma$, $\Phi$ holds

3. $A$ know that there is $\sigma$ such that, for every execution of $\sigma$, $\Phi$ holds

4. There is $\sigma$ such that $A$ know that, for every execution of $\sigma$, $\Phi$ holds
Case [4]: knowing how to play

- Single agent case: we take into account the paths starting from indistinguishable states (i.e., $\bigcup_{q' \in \text{img}_q, \sim_a} \text{out}(q, s_A)$)

- What about coalitions?
- Question: in what sense should they know the strategy? Common knowledge ($C_A$), mutual knowledge ($K_A$), distributed knowledge ($D_A$)?
Given strategy $\sigma$, agents $A$ can have:

- **Common knowledge** that $\sigma$ is a winning strategy. This requires the least amount of additional communication (agents from $A$ may agree upon a total order over their collective strategies at the beginning of the game and that they will always choose the maximal winning strategy with respect to this order).

- **Mutual knowledge** that $\sigma$ is a winning strategy: everybody in $A$ knows that $\sigma$ is winning.
3. Reasoning about Coalitions

- **Distributed knowledge** that \( \sigma \) is a winning strategy: if the agents share their knowledge at the current state, they can identify the strategy as winning.

- **“The leader”**: the strategy can be identified by agent \( a \in A \).

- **“Headquarters’ committee”**: the strategy can be identified by subgroup \( A' \subseteq A \).

- **“Consulting company”**: the strategy can be identified by some other group \( B \).
Many subtle cases...

Solution: constructive knowledge operators
Constructive Strategic Logic (CSL)

- $\langle A \rangle \Phi$: $A$ have a uniform memoryless strategy to enforce $\Phi$
- $K_a \langle a \rangle \Phi$: $a$ has a strategy to enforce $\Phi$, and knows that he has one
- For groups of agents: $C_A, E_A, D_A, ...$
- $\mathbb{K}_a \langle a \rangle \Phi$: $a$ has a strategy to enforce $\Phi$, and knows that this is a winning strategy
- For groups of agents: $C_A, E_A, D_A, ...$
Non-standard semantics:

- Formulae are evaluated in sets of states.
- \( M, Q \models \langle A \rangle \Phi \): A have a \textbf{single} strategy to enforce \( \Phi \) from all states in \( Q \)

Additionally:

- \( out(Q, S_A) = \bigcup_{q \in Q} out(q, S_A) \)
- \( img(Q, R) = \bigcup_{q \in Q} img(q, R) \)
- \( M, q \models \varphi \) iff \( M, \{ q \} \models \varphi \)
Definition 3.7 (Semantics of CSL)

\[ M, Q \models p \iff p \in \pi(q) \text{ for every } q \in Q; \]
\[ M, Q \models \neg \varphi \iff \text{not } M, Q \models \varphi; \]
\[ M, Q \models \varphi \land \psi \iff M, Q \models \varphi \text{ and } M, Q \models \psi; \]
\[ M, Q \models \langle A \rangle_{\gamma} \iff \text{there exists } S_A \text{ such that, for every } \lambda \in \text{out}(Q, S_A), \text{ we have that } M, \lambda[1] \models \varphi; \]
\[ M, Q \models \mathcal{K}_A \varphi \text{ iff } M, q \models \varphi \text{ for every } q \in \text{img}(Q, \sim \mathcal{K}_A) \text{ (where } \mathcal{K} = C, E, D); \]

\[ M, Q \models \hat{\mathcal{K}}_A \varphi \text{ iff } M, \text{img}(Q, \sim \hat{\mathcal{K}_A}) \models \varphi \text{ (where } \hat{\mathcal{K}} = C, E, D \text{ and } \mathcal{K} = C, E, D, \text{ respectively}). \]
Example: Simple Market

\[ @ \ q_1 : \]
\[ \neg K_c \langle c \rangle \diamond \text{success} \]
\[ \neg E_{\{1,2\}} \langle c \rangle \diamond \text{success} \]
\[ \neg K_1 \langle c \rangle \diamond \text{success} \]
\[ \neg K_2 \langle c \rangle \diamond \text{success} \]
\[ \mathcal{D}_{\{1,2\}} \langle c \rangle \diamond \text{success} \]
Theorem 3.8 (Expressivity)

**CSL is strictly more expressive than most previous proposals.**

Theorem 3.9 (Verification complexity)

**The complexity of model checking CSL is minimal.**
3.5 Model Checking
Model Checking Formulae of CTL and ATL

- Model checking: Does $\varphi$ hold in model $M$ and state $q$?

- Natural for verification of existing systems; also during design (“prototyping”)

- Can be used for automated planning
3. Reasoning about Coalitions

function \text{plan}(\varphi).  
Returns a subset of \(Q\) for which formula \(\varphi\) holds, together with a (conditional) plan to achieve \(\varphi\). The plan is sought within the context of concurrent game structure \(S = \langle \text{Agt}, Q, \Pi, \pi, o \rangle\).

<table>
<thead>
<tr>
<th>Case</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\varphi \in \Pi)</td>
<td>return ({\langle q, - \rangle \mid \varphi \in \pi(q)})</td>
</tr>
<tr>
<td>(\varphi = \neg \psi)</td>
<td>(P_1 := \text{plan}(\psi)); return ({\langle q, - \rangle \mid q \notin \text{states}(P_1)})</td>
</tr>
<tr>
<td>(\varphi = \psi_1 \lor \psi_2)</td>
<td>(P_1 := \text{plan}(\psi_1); P_2 := \text{plan}(\psi_2)); return ({\langle q, - \rangle \mid q \in \text{states}(P_1) \cup \text{states}(P_2)})</td>
</tr>
<tr>
<td>(\varphi = \langle A \rangle \bigcirc \psi)</td>
<td>return (\text{pre}(A, \text{states}(\text{plan}(\psi))))</td>
</tr>
</tbody>
</table>
| \(\varphi = \langle A \rangle \Box \psi\) | \(P_1 := \text{plan}(\text{true}); P_2 := \text{plan}(\psi); Q_3 := \text{states}(P_2)\);  
while \(\text{states}(P_1) \not\subseteq \text{states}(P_2)\)  
do \(P_1 := P_2|_{\text{states}(P_1)}; P_2 := \text{pre}(A, \text{states}(P_1))|_{Q_3}\)  
end while  
return \(P_2|_{\text{states}(P_1)}\) |
| \(\varphi = \langle A \rangle \psi_1 \bigcup \psi_2\) | \(P_1 := \emptyset; Q_3 := \text{states}(\text{plan}(\psi_1)); P_2 := \text{plan}(\text{true})|_{\text{states}(\text{plan}(\psi_2))}\);  
while \(\text{states}(P_2) \not\subseteq \text{states}(P_1)\)  
do \(P_1 := P_1 \oplus P_2; P_2 := \text{pre}(A, \text{states}(P_1))|_{Q_3}\)  
end while  
return \(P_1\) |

5. Model Checking
Complexity of Model Checking ATL

Theorem (Alur, Kupferman & Henzinger 1998)

ATL model checking is $P$-complete, and can be done in time linear in the size of the model and the length of the formula.

So, let’s model-check!

Not as easy as it seems.
Nice results: model checking ATL is tractable.

But: the result is relative to the size of the model and the formula

Well known catch: size of models is exponential wrt a higher-level description

Another problem: transitions are labeled

So: the number of transitions can be exponential in the number of agents.
3 agents/attributes, 12 states, 216 transitions
### Model Checking Temporal & Strategic Logics

<table>
<thead>
<tr>
<th>Logic</th>
<th>$m, l$</th>
<th>$n, k, l$</th>
<th>$n_{local}, k, l$</th>
</tr>
</thead>
</table>

Main message:

- Complexity is very sensitive to the context!
- In particular, the way we define the input, and measure its size, is crucial.
Even if model checking appears very easy, it can be very hard.

Still, people do automatic model checking!

- LTL: SPIN
- CTL/ATL: MOCHA, MCMAS, VeriCS

Even if model checking is theoretically hard, it can be feasible in practice.
3. Reasoning about Coalitions

6. References


