Modal Logics for Games and Multi-Agent Systems

Thomas Ågotnes and Wojtek Jamroga
(Bergen University College / TU Clausthal)
Time and place:
11–15 August 2008, hours 9.15-10.45

Organization:  
Parts 2, 4, 7, 9 & 10:  
Thomas Ågotnes,  
Parts 1, 3, 5, 6 & 8:  
Wojtek Jamroga
Lecture Overview I


2. **Coalition logic.** Strategic games and coalition logic (CL). Axiomatisation of CL.

3. **ATL.** Multi-step games and alternating-time temporal logic (ATL).

4. **More about ATL.** Axiomatisation; bisimulation; the role of memory; revocability of strategies.
Lecture Overview II

5. **Strategic reasoning for imperfect information (part I)**. Strategic reasoning for imperfect information scenarios. Problems with ATEL. Economic solution: ATLir.


Lecture Overview III

8 **Reasoning about rational play.** Reasoning about rational play in ATLP. Temporalized solution concepts.


10 **Axiomatisation of coalitional games (part II).** Axiomatisation of coalitional games: completeness proof.
Basic Reading I


Basic Reading II


Agent Systems and Modal Logic
1. Agent Systems and Modal Logic

1.1 Agents
1. Agent Systems and Modal Logic

1. Agents

- **Multi-agent system (MAS):** a system that involves several *autonomous* entities that *act* in the same environment
- The entities are called *agents*
Multi-agent system (MAS): a system that involves several autonomous entities that act in the same environment.

The entities are called agents.

So, what is an agent precisely?
Multi-agent system (MAS): a system that involves several autonomous entities that act in the same environment

The entities are called agents

So, what is an agent precisely?

No commonly accepted definition
For some authors, agents are:

- A new paradigm for computation
For some authors, agents are:

- A new paradigm for computation
- A new paradigm for design
For some authors, agents are:

- A new paradigm for computation
- A new paradigm for design
- A new paradigm for programming
For some authors, agents are:

- A new paradigm for computation
- A new paradigm for design
- A new paradigm for programming

Our claim:

MAS is a philosophical metaphor that induces a specific way of seeing the world.
Intuition:

We are agents!
Intuition: We are agents!

The metaphor:
- Makes us use specific vocabulary
Intuition:
We are agents!

The metaphor:
- Makes us use specific vocabulary
- Makes us use specific conceptual structures
1. Agent Systems and Modal Logic

1. Agents

Intuition:
We are agents!

The metaphor:
- Makes us use specific vocabulary
- Makes us use specific conceptual structures
- So:
- A new paradigm for thinking and talking about the world
Features of agents

An agent can/should possibly be:
Features of agents

An agent can/should possibly be:

- **Autonomous:** operates without direct intervention of others, has some kind of control over its actions and internal state
Features of agents

An agent can/should possibly be:

- **Autonomous**: operates without direct intervention of others, has some kind of control over its actions and internal state
- **Reactive**: reacts to changes in the environment
Features of agents

An agent can/should possibly be:

- **Autonomous**: operates without direct intervention of others, has some kind of control over its actions and internal state
- **Reactive**: reacts to changes in the environment
- **Pro-active**: takes the initiative
Features of agents

An agent can/should possibly be:

- **Autonomous**: operates without direct intervention of others, has some kind of control over its actions and internal state
- **Reactive**: reacts to changes in the environment
- **Pro-active**: takes the initiative
- **Goal-directed**: acts to achieve a goal
Features of agents

An agent can/should possibly be:

- **Autonomous**: operates without direct intervention of others, has some kind of control over its actions and internal state
- **Reactive**: reacts to changes in the environment
- **Pro-active**: takes the initiative
- **Goal-directed**: acts to achieve a goal
- **Social**: interacts with others (cooperation, communication, coordination, competition)
1. Agent Systems and Modal Logic

1. Agents

Features of agents

- **Embodied**: has *sensors* and *effectors* to read from and make changes to the environment
Features of agents

- **Embodied:** has sensors and effectors to read from and make changes to the environment
- **Intelligent:**
Features of agents

- **Embodied**: has *sensors* and *effectors* to read from and make changes to the environment
- **Intelligent**: ...whatever it means
Features of agents

- **Embodied**: has sensors and effectors to read from and make changes to the environment
- **Intelligent**: ...whatever it means
- **Rational**: always does the right thing
Is there any essential (and commonly accepted) feature of an agent?
Is there any essential (and commonly accepted) feature of an agent?

An agent acts.
Is there any essential (and commonly accepted) feature of an agent?

An agent acts.

Agents can be described mathematically by a function

\[ \text{act} : \text{set of percept sequences} \mapsto \text{set of actions} \]
Is there any essential (and commonly accepted) feature of an agent?

An agent acts.

Agents can be described mathematically by a function

\[ \text{act} : \text{set of percept sequences} \rightarrow \text{set of actions} \]

Note that, in game theory, such a function is called a strategy.
Is there any essential (and commonly accepted) feature of an agent?

An agent *acts*.

Agents can be described mathematically by a function

\[
\text{act} : \text{set of percept sequences} \mapsto \text{set of actions}
\]

Note that, in game theory, such a function is called a *strategy*.

In planning, it is called a *conditional plan*.
1.2 Modal Logic
Modal logic is an extension of classical logic by new connectives $\square$ and $\Diamond$: necessity and possibility.
Modal logic

**Modal logic** is an extension of classical logic by new connectives $\Box$ and $\Diamond$: necessity and possibility.

- $\Box \varphi$ means that $\varphi$ is necessarily true
Modal logic

Modal logic is an extension of classical logic by new connectives $\Box$ and $\diamond$: necessity and possibility.

- $\Box \varphi$ means that $\varphi$ is necessarily true
- $\diamond \varphi$ means that $\varphi$ is possibly true
Modal logic

Modal logic is an extension of classical logic by new connectives $\Box$ and $\Diamond$: necessity and possibility.

- $\Box \varphi$ means that $\varphi$ is necessarily true
- $\Diamond \varphi$ means that $\varphi$ is possibly true

Independently of the precise definition, the following holds:

$$\Diamond \varphi \iff \neg \Box \neg \varphi$$
More precisely, necessity/possibility is interpreted as follows:

- $p$ is necessary $\iff p$ is true in all possible scenarios
- $p$ is possible $\iff p$ is true in at least one possible scenario
More precisely, necessity/possibility is interpreted as follows:

- $p$ is necessary $\iff p$ is true in all possible scenarios
- $p$ is possible $\iff p$ is true in at least one possible scenario

$\leadsto$ possible worlds semantics
Definition 1.1 (Kripke structure)

A Kripke structure is a tuple \( S = \langle W, R \rangle \), where \( W \) is a set of possible worlds, and \( R \) is a binary relation on worlds, called accessibility relation.

For multiple modalities \( \Box_1, \Diamond_1, \ldots, \Box_k, \Diamond_k \), we use a family of relations \( R_1, \ldots, R_k \).
1. Agent Systems and Modal Logic

Definition 1.1 (Kripke structure)

A Kripke structure is a tuple $S = \langle W, R \rangle$, where $W$ is a set of possible worlds, and $R$ is a binary relation on worlds, called accessibility relation.

For multiple modalities $\Box_1, \Diamond_1, \ldots, \Box_k, \Diamond_k$, we use a family of relations $R_1, \ldots, R_k$

Definition 1.2 (Kripke model)

Let $\Pi$ be a set of atomic propositions $(p, q, r, \ldots)$.

A possible worlds model $M = \langle S, \pi \rangle$ consists of a Kripke structure $S$, and a valuation of propositions $\pi : W \rightarrow 2^\Pi$. 
Definition 1.3 (Semantics of modal logic)

The truth of formulae is relative to a Kripke model $M = \langle W, R, \pi \rangle$, and a world $w \in W$. It can be defined through the following clauses:
Definition 1.3 (Semantics of modal logic)

The truth of formulae is relative to a Kripke model \( M = \langle \mathcal{W}, \mathcal{R}, \pi \rangle \), and a world \( w \in \mathcal{W} \). It can be defined through the following clauses:

- \( M, w \models p \iff p \in \pi(w) \);
Definition 1.3 (Semantics of modal logic)

The truth of formulae is relative to a Kripke model $M = \langle \mathcal{W}, \mathcal{R}, \pi \rangle$, and a world $w \in \mathcal{W}$. It can be defined through the following clauses:

- $M, w \models p$ iff $p \in \pi(w)$;
- $M, w \models \neg \varphi$ iff not $M, w \models \varphi$;
Definition 1.3 (Semantics of modal logic)

The truth of formulae is relative to a Kripke model $M = \langle W, R, \pi \rangle$, and a world $w \in W$. It can be defined through the following clauses:

- $M, w \models p$ iff $p \in \pi(w)$;
- $M, w \models \neg \varphi$ iff not $M, w \models \varphi$;
- $M, w \models \varphi \land \psi$ iff $M, w \models \varphi$ and $M, w \models \psi$;
Definition 1.3 (Semantics of modal logic)

The truth of formulae is relative to a Kripke model $M = \langle W, R, \pi \rangle$, and a world $w \in W$. It can be defined through the following clauses:

- $M, w \models p$ iff $p \in \pi(w)$;
- $M, w \models \neg \varphi$ iff not $M, w \models \varphi$;
- $M, w \models \varphi \land \psi$ iff $M, w \models \varphi$ and $M, w \models \psi$;
- $M, w \models \Box \varphi$ iff, for every $w' \in W$ such that $wRw'$, we have $M, w' \models \varphi$. 

Thomas Ågotnes and Wojtek Jamroga · Modal Logics for Games and MAS

ESSLLI 2008 @ Hamburg 20/355
Logic for Agents

Modal logic is a **generic** framework.
Logic for Agents

Modal logic is a **generic** framework.

Various modal logics:
- knowledge $\rightsquigarrow$ **epistemic logic**,
- beliefs $\rightsquigarrow$ **doxastic logic**,
- obligations $\rightsquigarrow$ **deontic logic**,
- actions $\rightsquigarrow$ **dynamic logic**,
- time $\rightsquigarrow$ **temporal logic**,
- ability $\rightsquigarrow$ **strategic logic**,
- and **combinations of the above**
Logic for Agents

Modal logic is a generic framework.

Various modal logics:

- knowledge $\rightsquigarrow$ epistemic logic,
- beliefs $\rightsquigarrow$ doxastic logic,
- obligations $\rightsquigarrow$ deontic logic,
- actions $\rightsquigarrow$ dynamic logic,
- time $\rightsquigarrow$ temporal logic,
- ability $\rightsquigarrow$ strategic logic,
- and combinations of the above

Modal logic seems very well suited for reasoning about various dimensions of multi-agent systems!
1.3 Epistemic Logic
Epistemic logic

■ We interpret $\Box_i \varphi$ as “agent $i$ knows that $\varphi$”
■ $\Box_i$ is usually written as $K_i$ in epistemic logic
Epistemic logic

- We interpret $\Box_i \varphi$ as “agent $i$ knows that $\varphi$”
- $\Box_i$ is usually written as $K_i$ in epistemic logic

- Possible worlds: states of the system, situations
- Modal relations $R_i$: indistinguishability of states for agent $i$
- We assume that $R_i$ are equivalence relations
Epistemic logic

- We interpret $\square_i \varphi$ as “agent $i$ knows that $\varphi$”
- $\square_i$ is usually written as $K_i$ in epistemic logic

- Possible worlds: states of the system, situations
- Modal relations $\sim_i$: indistinguishability of states for agent $i$
- We assume that $\sim_i$ are equivalence relations
Epistemic logic

- We interpret $\Box_i \varphi$ as “agent $i$ knows that $\varphi$”
- $\Box_i$ is usually written as $K_i$ in epistemic logic

Possible worlds: states of the system, situations

Modal relations $\sim_i$: indistinguishability of states for agent $i$

- We assume that $\sim_i$ are equivalence relations

$M, w \models K_i \varphi$ iff $\varphi$ holds in all worlds that look the same as $w$
Example: Robots and Carriage
Example: Robots and Carriage
Example: Robots and Carriage

\[ q_0 \]
\[ q_1 \]
\[ q_2 \]
\[ pos_0 \]
\[ pos_1 \]
\[ pos_2 \]
Example: Robots and Carriage
Example: Robots and Carriage
Example: Robots and Carriage
Example: Robots and Carriage

![Diagram of states q0, q1, q2 with transitions pos0, pos1, pos2]
Example: Robots and Carriage

\[ \text{pos}_2 \rightarrow \neg K_1 \text{pos}_2 \]
Example: Robots and Carriage

\[ \text{pos}_2 \rightarrow \neg K_1 \text{pos}_2 \]
\[ \text{pos}_2 \rightarrow K_1 \neg \text{pos}_1 \]
Example: Robots and Carriage

\[ pos_2 \rightarrow \neg K_1 pos_2 \]
\[ pos_2 \rightarrow K_1 \neg pos_1 \]
\[ pos_2 \rightarrow K_2 K_1 \neg pos_1 \]
Logical omniscience

If $\varphi$ is valid then $K_i \varphi$ also holds
Logical omniscience

If $\varphi$ is valid then $K_i\varphi$ also holds

Problem!
Logical omniscience

If $\varphi$ is valid then $K_i \varphi$ also holds

Problem!

Example

Do the whites have a winning strategy in chess?
Collective knowledge

A group of agents $A$ can “know” that $\varphi$ in several different epistemic modes:
Collective knowledge

A group of agents $A$ can “know” that $\varphi$ in several different epistemic modes:

- $E_A \varphi$: everybody in $A$ knows that $\varphi$ ($A$ have mutual knowledge that $\varphi$)
Collective knowledge

A group of agents $A$ can “know” that $\varphi$ in several different epistemic modes:

- $E_A\varphi$: everybody in $A$ knows that $\varphi$ ($A$ have mutual knowledge that $\varphi$)
- $C_A\varphi$: it is a common knowledge among $A$ that $\varphi$
Collective knowledge

A group of agents $A$ can “know” that $\varphi$ in several different epistemic modes:

- $E_A\varphi$: everybody in $A$ knows that $\varphi$ (A have mutual knowledge that $\varphi$)
- $C_A\varphi$: it is a common knowledge among $A$ that $\varphi$
- $D_A\varphi$: $A$ have distributed knowledge that $\varphi$
Collective knowledge

A group of agents $A$ can “know” that $\varphi$ in several different epistemic modes:

- $E_A \varphi$: everybody in $A$ knows that $\varphi$ ($A$ have mutual knowledge that $\varphi$)
- $C_A \varphi$: it is a common knowledge among $A$ that $\varphi$
- $D_A \varphi$: $A$ have distributed knowledge that $\varphi$

Multi-agent Epistemic Logic ($\text{MAEL}_n$): $K_n$ plus modalities for mutual, common, and distributed knowledge
Collective knowledge: semantics

\[ M, q \models E_A \varphi \quad \text{iff} \quad M, q' \models \varphi \quad \text{for every } q' \text{ such that } q \sim_{E_A} q', \]
where \[ \sim_{E_A} = \bigcup_{i \in A} \sim_i \]
Collective knowledge: semantics

- $M, q \models E_A \varphi$ iff $M, q' \models \varphi$ for every $q'$ such that $q \sim_{E_A}^E q'$, where $\sim_{E_A}^E = \bigcup_{i \in A} \sim_i$

- $M, q \models C_A \varphi$ iff $M, q' \models \varphi$ for every $q'$ such that $q \sim_{C_A}^C q'$, where $\sim_{C_A}^C$ is the transitive closure of $\sim_{E_A}^E$
Collective knowledge: semantics

- \( M, q \models E_A \varphi \) iff \( M, q' \models \varphi \) for every \( q' \) such that \( q \sim_{E_A} q' \), where \( \sim_{E_A} = \bigcup_{i \in A} \sim_i \)

- \( M, q \models C_A \varphi \) iff \( M, q' \models \varphi \) for every \( q' \) such that \( q \sim_{C_A} q' \), where \( \sim_{C_A} \) is the transitive closure of \( \sim_{E_A} \)

- \( M, q \models D_A \varphi \) iff \( M, q' \models \varphi \) for every \( q' \) such that \( q \sim_{E_A} q' \), where \( \sim_{E_A} = \bigcap_{i \in A} \sim_i \)
1.4 Axioms
As in classical logic, one can ask about a complete axiom system. Is there a calculus that allows to derive all sentences that are true in all Kripke models?
As in classical logic, one can ask about a complete axiom system. Is there a calculus that allows to derive all sentences that are true in all Kripke models?

**Definition 1.4 (System K)**

System K is an extension of the propositional calculus by the axiom

\[ K (\Box \varphi \land \Box (\varphi \rightarrow \psi)) \rightarrow \Box \psi \]

and the inference rule

\[ (\text{Necessitation}) \quad \frac{\varphi}{\Box \varphi} \]
Theorem 1.5 (Soundness/completeness of system $K$)

System $K$ is sound and complete with respect to the class of all Kripke models.
Theorem 1.5 (Soundness/completeness of system K)

System $K$ is sound and complete with respect to the class of all Kripke models.

Note: with $n$ modalities, the calculus is called $K_n$, and the theorem extends in a straightforward way.
Definition 1.6 (Extending K with axioms D, T, 4, 5)

System K is often extended by the following axioms:

\[
\begin{align*}
K & : (\Box \varphi \land \Box (\varphi \rightarrow \psi)) \rightarrow \Box \psi \\
D & : \neg \Box (\varphi \land \neg \varphi) \\
T & : \Box \varphi \rightarrow \varphi \\
4 & : \Box \varphi \rightarrow \Box \Box \varphi \\
5 & : \neg \Box \varphi \rightarrow \Box \neg \Box \varphi
\end{align*}
\]
Definition 1.6 (Extending K with axioms D, T, 4, 5)

System \( K \) is often extended by the following axioms:

- **K** \( (\Box \varphi \land \Box(\varphi \rightarrow \psi)) \rightarrow \Box \psi \)
- **D** \( \neg \Box(\varphi \land \neg \varphi) \)
- **T** \( \Box \varphi \rightarrow \varphi \)
- **4** \( \Box \varphi \rightarrow \Box \Box \varphi \)
- **5** \( \neg \Box \varphi \rightarrow \Box \neg \Box \varphi \)

Best known extensions of system \( K \):

- **S5 = KDT45**: the standard logic of **knowledge**
- **KD45**: the standard logic of **beliefs**
Theorem 1.7 (Sound/complete subsystems of KDT45)

Let $X$ be any subset of \{D, T, 4, 5\} and let $\mathcal{X}$ be any subset of \{serial, reflexive, transitive, euclidean\} corresponding to $X$.

Then $K \cup X$ is sound and complete with respect to Kripke models the accessibility relation of which satisfies $\mathcal{X}$. 
**Theorem 1.7 (Sound/complete subsystems of KDT45)**

Let $X$ be any subset of $\{D, T, 4, 5\}$ and let $\mathcal{X}$ be any subset of $\{\text{serial, reflexive, transitive, euclidean}\}$ corresponding to $X$.

Then $K \cup X$ is sound and complete with respect to Kripke models the accessibility relation of which satisfies $\mathcal{X}$.

**Corollary 1.8**

System $S5$ is sound and complete with respect to Kripke models with equivalence accessibility relations.
Theorem 1.7 (Sound/complete subsystems of KDT45)

Let $X$ be any subset of $\{D, T, 4, 5\}$ and let $\mathcal{X}$ be any subset of $\{\text{serial, reflexive, transitive, euclidean}\}$ corresponding to $X$.

Then $K \cup X$ is sound and complete with respect to Kripke models the accessibility relation of which satisfies $\mathcal{X}$.

Corollary 1.8

System $S5$ is sound and complete with respect to Kripke models with equivalence accessibility relations.

Theorem 1.9

Deciding if $\varphi$ is a theorem of $S5$ is PSPACE-complete.
Axioms for collective knowledge

Axioms for MAEL\(_n\) extend S5\(_n\) with schemes:

- Axioms for \(E_A\): \(E_A \varphi \leftrightarrow \bigwedge_{i \in A} E_i \varphi\)
Axioms for collective knowledge

Axioms for \( \text{MAEL}_n \) extend \( \text{S5}_n \) with schemes:

- Axioms for \( E_A \): \( E_A \varphi \leftrightarrow \bigwedge_{i \in A} E_i \varphi \)
- Segerberg’s axioms for \( E_A \) and \( C_A \):
  
  \[
  \begin{align*}
  \text{MIX}_A & : \quad C_A \varphi \to (\varphi \land E_A C_A \varphi) \\
  \text{IND}_A & : \quad \varphi \land C_A (\varphi \to E_A \varphi) \to C_A \varphi
  \end{align*}
  \]
Axioms for collective knowledge

Axioms for MAEL\(_n\) extend S5\(_n\) with schemes:

- Axioms for \(E_A\):  
  \[ E_A \varphi \leftrightarrow \bigwedge_{i \in A} E_i \varphi \]

- Segerberg’s axioms for \(E_A\) and \(C_A\):
  \[
  \text{MIX}_A : \quad C_A \varphi \to (\varphi \land E_A C_A \varphi) \\
  \text{IND}_A : \quad \varphi \land C_A (\varphi \to E_A \varphi) \to C_A \varphi
  \]

- Axioms for \(D_A\):
  \[
  S5(D_A) : \quad \text{The S5 axioms for } D_A, \\
  D_i : \quad D_i \varphi \leftrightarrow E_i \varphi, \\
  \text{INCL}(D) : \quad D_A \varphi \to D_B \varphi \text{ whenever } A \subseteq B.
  \]
Axioms for collective knowledge

Axioms for \( \text{MAEL}_n \) extend \( S5_n \) with schemes:

- Axioms for \( E_A \):  \( E_A \varphi \leftrightarrow \bigwedge_{i \in A} E_i \varphi \)
- Segerberg’s axioms for \( E_A \) and \( C_A \):
  
  - \( \text{MIX}_A \):  \( C_A \varphi \rightarrow (\varphi \land E_A C_A \varphi) \)
  - \( \text{IND}_A \):  \( \varphi \land C_A (\varphi \rightarrow E_A \varphi) \rightarrow C_A \varphi \)

- Axioms for \( D_A \):
  
  - \( \text{S5}(D_A) \):  The \( S5 \) axioms for \( D_A \),
  - \( D_i \):  \( D_i \varphi \leftrightarrow E_i \varphi \),
  - \( \text{INCL}(D) \):  \( D_A \varphi \rightarrow D_B \varphi \) whenever \( A \subseteq B \).

- ...plus, for each \( i \in \text{Agt} \), the inference rule
  
  \[
  \frac{\varphi}{E_i \varphi}
  \]
Theorem 1.10 (Soundness/completeness of MAEL$_n$)

The axiom system for MAEL$_n$ is sound and complete.
Theorem 1.10 (Soundness/completeness of MAEL<sub>n</sub>)

The axiom system for MAEL<sub>n</sub> is sound and complete.

Theorem 1.11

Deciding if ϕ is a theorem of MAEL<sub>n</sub> is EXPTIME-complete. It remains EXPTIME-complete even if only common or distributed knowledge operators are included.
Multi-agent systems:


*An Introduction to Multi Agent Systems.*  

FIPA home page.  
http://www.fipa.org/.
1. Agent Systems and Modal Logic

Modal logic:


Reasoning about knowledge:

Reasoning about Knowledge.

Reasoning about knowledge: a survey. 

Epistemic logic: A survey. 
2. Coalition Logic

2.1 Strategic Games
Game Theory

- Game theory is concerned with understanding what happens when rational decision-makers interact.

- Non-cooperative game theory: actions of individual players are taken as primitives.
  - Strategic games: players simultaneously choose complete strategies at the beginning of the game.
  - Extensive games: players can postpone decisions about which actions to choose until they are needed.

- Cooperative (or coalitional) game theory: actions of coalitions, i.e. groups of players, are taken as primitives.
Game Theory

- Game theory is concerned with understanding what happens when rational decision-makers interact.
- **Non-cooperative** game theory: actions of individual players are taken as primitives.
  - **Strategic games**: players simultaneously choose complete strategies at the beginning of the game.
  - **Extensive games**: players can postpone decisions about which actions to choose until they are needed.
- **Cooperative (or coalitional)** game theory: actions of coalitions, i.e. groups of players, are taken as primitives.

This lecture: (repeated) strategic game forms. (Lecture 3: extensive form games; lectures 7 and 8: solution concepts for non-cooperative games; lectures 9 and 10: solution concepts for coalitional games)
Strategic game form

A game form is a tuple

\[ G = (N, \{\Sigma_i : i \in N\}, o, S) \]

where

- \( N \) is a nonempty finite set of players
- \( S \) is a nonempty set of states (or outcomes or consequences)
- \( \Sigma_i \) is the nonempty set of actions (strategies) for agent \( i \)
- \( o : \times_{i \in N} \Sigma_i \rightarrow S \) associates a state with a strategy profile
Strategic game form

A **game form** is a tuple

\[ G = (N, \{\Sigma_i : i \in N\}, o, S) \]

where

- \( N \) is a nonempty finite set of **players**
- \( S \) is a nonempty set of **states** (or **outcomes** or **consequences**)
- \( \Sigma_i \) is the nonempty set of **actions** (**strategies**) for agent \( i \)
- \( o : \times_{i \in N} \Sigma_i \rightarrow S \) associates a **state** with a **strategy profile**

**Notation:**

- \( \sigma_G \) where \( G \subseteq N \): a partial strategy profile \( \sigma_G \subseteq \times_{i \in G} \Sigma_i \)
- \( \sigma_{-i} : \) same as \( \sigma_{N\setminus\{i\}} \)
- \( \Gamma^N_S \) is the set of all game forms with players \( N \) and
Strategic game

When

\[ G = (N, \{\Sigma_i : i \in N\}, o, S) \]

and, for each player \( i \in N \),

\[ \succeq_i \subseteq S \times S \]

is a preference relation (complete, reflexive, transitive),
then

\[ (G, \{\succeq_i : i \in N\}) \]

is a strategic game
Example: Prisoners’ Dilemma

<table>
<thead>
<tr>
<th></th>
<th>Bill</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cooperate</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ann</td>
<td>Cooperate</td>
<td>Ann:-1, Bill:-1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Defect</td>
<td>Ann:0, Bill:-4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cooperate</td>
<td>Ann:-4, Bill: 0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Defect</td>
<td>Ann:-3, Bill: -3</td>
<td></td>
</tr>
</tbody>
</table>
2. Coalition Logic

Example: Prisoners’ Dilemma

<table>
<thead>
<tr>
<th></th>
<th>Cooperate</th>
<th>Defect</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ann</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cooperate</td>
<td>Ann:-1, Bill:-1</td>
<td>Ann:-4, Bill: 0</td>
</tr>
<tr>
<td>Defect</td>
<td>Ann:0, Bill:-4</td>
<td>Ann:-3, Bill: -3</td>
</tr>
</tbody>
</table>

On game form:

\[ PD = (N, \{\Sigma_i : i \in N\}, o, S) \]

where

- \( N = \{Ann, Bill\} \)
- \( \Sigma_{Ann} = \Sigma_{Bill} = \{C, D\} \)
- \( S = \{-1/-1, -4/0, 0/-4, -3/-3\} \)
- \( o(C, C) = -1/-1, o(C, D) = -4/0, \text{ etc.} \)
2. Coalition Logic

2.2 Coalition Logic: Introduction
Modal Logic and Games

- (Extensive form) games look like Kripke structures! (van Benthem)
- Here we use strategic games and Marc Paulys Coalition Logic as a starting point
Coalition Logic (Pauly 2001)

- We can interpret modal logic in transition systems, and reason about how the system possibly or necessarily will evolve.
- **Game frames**: transition systems where the transitions are determined by a strategic game form in each state (and where the outcomes are, again, states).
- **Coalition logic**: about what coalitions can do (or ensure or make come about) – coalitional power.
- Main construct, $C \subseteq N$:
  \[\langle \langle C \rangle \rangle \varphi\]
  $C$ can make $\varphi$ come about.

- Alternative interpretations:
  - A logic of game frames
  - A logic of coalitional effectivity
Example 2.1

Two individuals, $A$ and $B$, must choose between two outcomes, $p$ and $q$. We want a mechanism that will allow them to choose which will satisfy the following requirements: We want an outcome to be possible – that is, we want the two agents to choose, collectively, either $p$ or $q$. We do not want them to be able to bring about both outcomes simultaneously. Finally, we do not want either agent to be able to unilaterally dictate an outcome – we want them both to have “equal power”.

\[\langle \langle A, B \rangle \rangle p \langle \langle A, B \rangle \rangle q \neg \langle \langle A \rangle \rangle p \neg \langle \langle B \rangle \rangle q\]

\[\neg \langle \langle A \rangle \rangle (p \land q) \neg \langle \langle B \rangle \rangle (p \land q)\]
Example 2.1

Two individuals, $A$ and $B$, must choose between two outcomes, $p$ and $q$. We want a mechanism that will allow them to choose which will satisfy the following requirements: We want an outcome to be possible – that is, we want the two agents to choose, collectively, either $p$ or $q$. We do not want them to be able to bring about both outcomes simultaneously. Finally, we do not want either agent to be able to unilaterally dictate an outcome – we want them both to have “equal power”.

These requirements may be formally and naturally represented using Coalition Logic, as follows:
Example 2.1

Two individuals, $A$ and $B$, must choose between two outcomes, $p$ and $q$. We want a mechanism that will allow them to choose which will satisfy the following requirements: We want an outcome to be possible – that is, we want the two agents to choose, collectively, either $p$ or $q$. We do not want them to be able to bring about both outcomes simultaneously. Finally, we do not want either agent to be able to unilaterally dictate an outcome – we want them both to have “equal power”.

These requirements may be formally and naturally represented using Coalition Logic, as follows:

$$\langle \langle A, B \rangle \rangle p \quad \langle \langle A, B \rangle \rangle q$$
Example 2.1

Two individuals, $A$ and $B$, must choose between two outcomes, $p$ and $q$. We want a mechanism that will allow them to choose which will satisfy the following requirements: We want an outcome to be possible – that is, we want the two agents to choose, collectively, either $p$ or $q$. We do not want them to be able to bring about both outcomes simultaneously. Finally, we do not want either agent to be able to unilaterally dictate an outcome – we want them both to have “equal power”.

These requirements may be formally and naturally represented using Coalition Logic, as follows:

$$\langle \langle A, B \rangle \rangle p \quad \langle \langle A, B \rangle \rangle q \quad \neg \langle \langle A, B \rangle \rangle (p \land q)$$
Example 2.1

Two individuals, $A$ and $B$, must choose between two outcomes, $p$ and $q$. We want a mechanism that will allow them to choose which will satisfy the following requirements: We want an outcome to be possible – that is, we want the two agents to choose, collectively, either $p$ or $q$. We do not want them to be able to bring about both outcomes simultaneously. Finally, we do not want either agent to be able to unilaterally dictate an outcome – we want them both to have “equal power”.

These requirements may be formally and naturally represented using Coalition Logic, as follows:

$$\langle \langle A, B \rangle \rangle p \quad \langle \langle A, B \rangle \rangle q \quad \neg \langle \langle A, B \rangle \rangle (p \land q)$$
$$\neg \langle \langle A \rangle \rangle q \quad \neg \langle \langle A \rangle \rangle p \quad \neg \langle \langle B \rangle \rangle p$$
$$\neg \langle \langle B \rangle \rangle q$$
2.3 From Game Forms to Effectivity Functions
We want to reason about coalitional power: can a coalition $C$ achieve an outcome $X \subseteq S$?

Coalitional power can be explicitly formalised by effectivity functions.
We want to reason about coalitional power: can a coalition $C$ achieve an outcome $X \subseteq S$?

Coalitional power can be explicitly formalised by effectivity functions.

An effectivity function $E$ is a function:

$$E : \wp(N) \rightarrow \wp(\wp(S))$$

$X \in E(C)$ means that the coalition $C$ is effective for the outcome $X$.

(Note: the literature differs in conditions imposed on eff. functions. More about this later.)
A game form $G$ induces an effectivity function $E_G$:

$$X \in E_G(C) \iff \exists \sigma_C \forall \sigma_{N\setminus C} (\sigma_C, \sigma_{N\setminus C}) \in X$$

(where $\sigma_C$ is a tuple of strategies for $C$)
A game form $G$ induces an effectivity function $E_G$:

$$X \in E_G(C) \iff \exists \sigma_C \forall \sigma_{N\setminus C} \circ (\sigma_C, \sigma_{N\setminus C}) \in X$$

(where $\sigma_C$ is a tuple of strategies for $C$)

- This form of effectivity is called $\alpha$-effectivity
- $C$ is effective for $X \subseteq S$ iff $C$ has a strategy such that the next state will be in the set $X$, no matter which strategies the players in $N \setminus C$ use
Example

<table>
<thead>
<tr>
<th></th>
<th>Cooperate</th>
<th>Defect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ann</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cooperate</td>
<td>Ann:-1, Bill:-1</td>
<td>Ann:-4, Bill: 0</td>
</tr>
<tr>
<td>Defect</td>
<td>Ann:0, Bill:-4</td>
<td>Ann:-3, Bill: -3</td>
</tr>
</tbody>
</table>

$PD$ induces the effectivity function $E_{PD}$:

\[
E_{PD}(\emptyset) = \{ S \} \\
E_{PD}(\{Ann\}) = \{ \{ -1/ -1, -4/0 \}, \{ 0/ -4, -3/ -3 \} \}^+ \\
E_{PD}(\{Bill\}) = \{ \{ -1/ -1, 0/ -4 \}, \{ -4/0, -3/ -3 \} \}^+ \\
E_{PD}(\{Ann, Bill\}) = \wp(S) \setminus \emptyset
\]

where $X^+$ is $X$ closed under outcome-monotonicity (i.e.: $X \subseteq X^+$, and if $Y \in X^+$ and $Y \subseteq Y' \subseteq S$ then $Y' \in X^+$)
2.4 Coalition Logic
Syntax

\[ \phi ::= \]

\[ \bot \] contradiction

\[ p \] atomic prop.

\[ \neg \phi \] negation

\[ \phi \lor \phi \] disjunction

\[ \langle \langle C \rangle \rangle \phi \] \( C \) can enforce \( \phi \)

where \( C \subseteq N \)
Game Frames

A game frame is a pair

\[(S, \gamma)\]

where \(S\) are the states and

\[\gamma : S \rightarrow \Gamma_S^N\]

associates a strategic game form to each state
2. Coalition Logic

Game Frame Example: Repeated PD

Recall our game form: $PD = (N, \{\Sigma_i : i \in N\}, o, S)$ where

- $N = \{Ann, Bill\}$
- $\Sigma_{Ann} = \Sigma_{Bill} = \{C, D\}$
- $S = \{-1/-1, -4/0, 0/-4, -3/-3\}$
- $o(C, C) = -1/-1, o(C, D) = -4/0$, etc.
Game Frame Example: Repeated PD

Recall our game form: $PD = (N, \{\Sigma_i : i \in N\}, o, S)$ where

- $N = \{Ann, Bill\}$
- $\Sigma_{Ann} = \Sigma_{Bill} = \{C, D\}$
- $S = \{-1/ -1, -4/0, 0/ -4, -3/ -3\}$
- $o(C, C) = -1/ -1, o(C, D) = -4/0$, etc.

The game frame

$$\mathcal{F}_{PD} = (S, \gamma)$$

where

$$S = \{-1/ -1, -4/0, 0/ -4, -3/ -3\}$$

$$\gamma(s) = PD \text{ for all } s \in S$$

models repeated play of prisoners’ dilemma
Models

- A **model** is a pair 
  \[ \mathcal{M} = (\mathcal{F}, V) \]

where
  - \( \mathcal{F} \) is a game frame
  - \( V \) assigns propositional atoms to states
Interpretation

Truth of a formula in a state $s$ of a model $\mathcal{M}$:

- $\mathcal{M}, s \notmodels \bot$
- $\mathcal{M}, s \models p \iff s \in V(p)$ ($p$ atomic prop.)
- $\mathcal{M}, s \models \neg \phi \iff \mathcal{M}, s \notmodels \phi$
- $\mathcal{M}, s \models \phi \lor \psi \iff \mathcal{M}, s \models \phi$ or $\mathcal{M}, s \models \psi$
- $\mathcal{M}, s \models \langle \langle C \rangle \rangle \phi \iff \phi^\mathcal{M} \in E_{\gamma(s)}(C)$

where $\phi^\mathcal{M} = \{ s \in S : \mathcal{M}, s \models \phi \}$
Interpretation: Example

\[ M_{PD} = ((S, \gamma), V) \]

where

\[ S = \{1/1, 4/0, 0/4, 3/3\} \quad \gamma(s) = PD \text{ for each } s \in S \]

\( V \) assigns truth values to propositions of the type

- \( A = 1 \) (Ann gets one year)
- \( B \geq 3 \) (Bill gets at least three years)

in the natural way (e.g., \( A = 1 \in V(1/1) \))
Interpretation: Example

\[ \mathcal{M}_{PD} = ((S, \gamma), V) \]

where

\[ S = \{1/1, 4/0, 0/4, 3/3\} \quad \gamma(s) = PD \text{ for each } s \in S \]

\(V\) assigns truth values to propositions of the type

- \(A = 1\) (Ann gets one year)
- \(B \geq 3\) (Bill gets at least three years)

in the natural way (e.g., \(A = 1 \in V(1/1)\))

Let \(s\) be any state.

- \(\mathcal{M}_{PD}, s \models \langle \langle Ann \rangle \rangle B \geq 3\)
Interpretation: Example

\[ M_{PD} = ((S, \gamma), V) \]

where

\[ S = \{1/1, 4/0, 0/4, 3/3\} \quad \gamma(s) = PD \text{ for each } s \in S \]

\( V \) assigns truth values to propositions of the type

- \( A = 1 \) (Ann gets one year)
- \( B \geq 3 \) (Bill gets at least three years)

in the natural way (e.g., \( A = 1 \in V(1/1) \))

Let \( s \) be any state.

- \( M_{PD}, s \models \langle Ann \rangle B \geq 3 \)
- \( M_{PD}, s \models \langle \{Ann, Bill\} \rangle A = 1 \land B = 1 \)
2. Coalition Logic

Interpretation: Example

\[ M_{PD} = ((S, \gamma), V) \]

where

\[ S = \{1/1, 4/0, 0/4, 3/3\} \quad \gamma(s) = PD \text{ for each } s \in S \]

\( V \) assigns truth values to propositions of the type

- \( A = 1 \) (Ann gets one year)
- \( B \geq 3 \) (Bill gets at least three years)

in the natural way (e.g., \( A = 1 \in V(1/1) \))

Let \( s \) be any state.

- \( M_{PD}, s \models \langle \langle Ann \rangle \rangle B \geq 3 \)
- \( M_{PD}, s \models \langle \{Ann, Bill\} \rangle A = 1 \land B = 1 \)
- \( M_{PD}, s \models \neg \langle \langle Ann \rangle \rangle A = 0 \)
2. Coalition Logic

5. Effectivity Properties

Playable Effectivity Functions

An effectivity function

\[ E : \wp(N) \rightarrow \wp(\wp(S)) \]

is playable iff:

1. \( \forall C \subseteq N : \emptyset \notin E(C) \)
2. \( \forall C \subseteq N : S \in E(C) \)
3. \( \forall X \subseteq S : S \setminus X \notin E(\emptyset) \Rightarrow X \in E(N) \) (N-maximality)
4. \( \forall C : \forall X \subseteq X' \subseteq S : X \in E(C) \Rightarrow X' \in E(C) \) (outcome-monotonicity)
5. \( \forall C_1 \subseteq N : \forall C_2 \subseteq N : \forall X_1 \subseteq S : \forall X_2 \subseteq S : (C_1 \cap C_2 = \emptyset \text{ and } X_1 \in E(C_1) \text{ and } X_2 \in E(C_2)) \Rightarrow X_1 \cap X_2 \in E(C_1 \cup C_2) \) (superadditivity)
2. Coalition Logic

5. Effectivity Properties

Playable Effectivity Functions

An effectivity function

\[ E : \wp(N) \rightarrow \wp(\wp(S)) \]

is playable iff:

1. \( \forall C \subseteq N : \emptyset \notin E(C) \)
2. Coalition Logic

5. Effectivity Properties

Playable Effectivity Functions

An effectivity function

\[ E : \wp(N) \rightarrow \wp(\wp(S)) \]

is playable iff:

1. \( \forall C \subseteq N : \emptyset \notin E(C) \)
2. \( \forall C \subseteq N : S \in E(C) \)
Playable Effectivity Functions

An effectivity function

\[ E : \wp(N) \rightarrow \wp(\wp(S)) \]

is **playable** iff:

1. \( \forall C \subseteq N : \emptyset \notin E(C) \)
2. \( \forall C \subseteq N : S \in E(C) \)
3. \( \forall X \subseteq S : S \setminus X \notin E(\emptyset) \Rightarrow X \in E(N) \) (**\(N\)-maximality)**
Playable Effectivity Functions

An effectivity function

\[ E : \emptyset(N) \rightarrow \emptyset(\emptyset(S)) \]

is **playable** iff:

1. \( \forall C \subseteq N : \emptyset \notin E(C) \)
2. \( \forall C \subseteq N : S \in E(C) \)
3. \( \forall X \subseteq S : S \setminus X \notin E(\emptyset) \Rightarrow X \in E(N) \) (**N-maximality**)
4. \( \forall C : \forall X \subseteq X' \subseteq S : X \in E(C) \Rightarrow X' \in E(C) \)
   (**outcome-monotonicity**)
Playable Effectivity Functions

An effectivity function

\[ E : \wp(N) \to \wp(\wp(S)) \]

is **playable** iff:

1. \( \forall C \subseteq N: \emptyset \notin E(C) \)
2. \( \forall C \subseteq N: S \in E(C) \)
3. \( \forall X \subseteq S: S \setminus X \notin E(\emptyset) \Rightarrow X \in E(N) \) (\( N \)-maximality)
4. \( \forall C: \forall X \subseteq X' \subseteq S: X \in E(C) \Rightarrow X' \in E(C) \) (outcome-monotonicity)
5. \( \forall C_1 \subseteq N: \forall C_2 \subseteq N: \forall X_1 \subseteq S: \forall X_2 \subseteq S: (C_1 \cap C_2 = \emptyset \text{ and } X_1 \in E(C_1) \text{ and } X_2 \in E(C_2)) \Rightarrow X_1 \cap X_2 \in E(C_1 \cup C_2) \) (superadditivity)
2. Coalition Logic

Strategic Game Forms vs. Effectivity Functions

Q: which effectivity functions are induced by strategic game forms?
A: exactly the playable effectivity functions

Theorem 2.2 (Pauly)

An effectivity function is playable iff it is induced by some strategic game form
Since the induced effectivity function is the only information about the frame used in the interpretation, this means that coalition logic can be seen as a logic of playable effectivity functions.

Alternative – and equivalent – models:

\[ \mathcal{M} = ((S, E), V) \]

where \( E(s) \) associates a playable effectivity function to each state.

This is a neighbourhood semantics: we get a neighbourhood relation for each coalition.

Technically easier to work with.
Coalition Logic: Axiomatisation

A sound and complete axiomatisation of all models (ref. playability properties):

\[
\neg \langle C \rangle \bot \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \;
\]

\[
\neg \langle \emptyset \rangle \neg \phi \rightarrow \langle N \rangle \phi \quad (N)
\]

\[
\langle C \rangle (\phi \land \psi) \rightarrow \langle C \rangle \psi \quad (M)
\]

\[
\langle C_1 \rangle \phi_1 \land \langle C_2 \rangle \phi_2 \rightarrow \langle C_1 \cup C_2 \rangle (\phi_1 \land \phi_2)
\]

\[
\text{when } C_1 \cap C_2 = \emptyset \quad (S)
\]

\[
\phi, \phi \rightarrow \psi \quad (MP)
\]

\[
\psi \rightarrow \phi \leftrightarrow \psi \quad (EQ)
\]

\[
\langle C \rangle \phi \leftrightarrow \langle C \rangle \psi
\]

Thomas Ågotnes and Wojtek Jamroga · Modal Logics for Games and MAS

ESSLLI 2008 @ Hamburg 66/355
2. Coalition Logic

5. Effectivity Properties

Playability properties again

1. ∀C ⊆ N: ∅ ∉ E(C')
2. ∀C ⊆ N: S ∈ E(C)
3. ∀X ⊆ S: S \ X ∉ E(∅) ⇒ X ∈ E(N) (N-maximality)
4. ∀C: ∀X ⊆ X' ⊆ S: X ∈ E(C') ⇒ X' ∈ E(C') (outcome-monotonicity)
5. ∀C_1 ⊆ N: ∀C_2 ⊆ N: ∀X_1 ⊆ S: ∀X_2 ⊆ S: (C_1 ∩ C_2 = ∅ and X_1 ∈ E(C_1) and X_2 ∈ E(C_2))
⇒ X_1 ∩ X_2 ∈ E(C_1 ∪ C_2) (superadditivity)
Completeness

- Can be shown by an *canonical model* construction
- Standard Lindenbaum argument: every consistent set of formulae can be extended to a max. cons. set
Canonical Model

\[ M^c = ((S^c, E^c), V^c) \]

\( S^c \): all MCSs
Canonical Model

\[ \mathcal{M}^c = ((S^c, E^c), V^c) \]

- \( S^c \): all MCSs
- \( V^c \): \( s \in V^c(p) \iff p \in s \)
2. Coalition Logic

5. Effectivity Properties

Canonical Model

\[ \mathcal{M}^c = ((S^c, E^c), V^c) \]

- \( S^c \): all MCSs
- \( V^c \): \( s \in V^c(p) \iff p \in s \)
- \( E^c \): associates with each \( s \) the canonical effectivity function \( E^c(s) \):

\[
X \in E^c(s)(G) \iff \\
\begin{cases} \\
\exists \phi : \tilde{\phi} \subseteq X \text{ and } \langle G \rangle \phi \in s & G \neq N \\
\forall \phi : \text{ if } \tilde{\phi} \subseteq S^c \setminus X \text{ then } \langle \emptyset \rangle \phi \notin s & G = N \\
\end{cases}
\]

where \( \tilde{\phi} = \{ s \in S^c : \phi \in s \} \)
Completeness

Truth Lemma

\[ M^c, s \models \phi \iff \phi \in s \]

for any MCS \( s \) and any \( \phi \)
Completeness

Truth Lemma

\[ M^c, s \models \phi \iff \phi \in s \]

for any MCS \( s \) and any \( \phi \)

- Easy to show by induction over \( \phi \), using the fact that we can derive the rule

\[
\phi \rightarrow \psi \\
\Rightarrow \langle \langle C' \rangle \rangle \phi \rightarrow \langle \langle C' \rangle \rangle \psi
\]
It remains to be shown that $E^c(s)$ is playable.
It remains to be shown that $E^c(s)$ is playable.

**Proposition: Alternative playability properties**

An effectivity function $E$ is playable iff:

1. $\forall C \neq N: \emptyset \not\in E(C)$
2. $\forall C \neq N: S \in E(C)$
3. $\forall X \subseteq S: S \setminus X \not\in E(\emptyset) \Rightarrow X \in E(N)$ ($N$-maximality)
4. $\forall C \neq N: \forall X \subseteq X' \subseteq S: X \in E(C) \Rightarrow X' \in E(C)$ (outcome-monotonicity)
5. $\forall C_1 \subset N: \forall C_2 \subset N$ s.t. $C_1 \cup C_2 \neq N: \forall X_1 \subseteq S: \forall X_2 \subseteq S: (C_1 \cap C_2 = \emptyset$ and $X_1 \in E(C_1)$ and $X_2 \in E(C_2)) \Rightarrow X_1 \cap X_2 \in E(C_1 \cup C_2)$ (superadditivity)
6. $\forall C \subseteq N: \forall X \subseteq S: \text{if } X \in E(C) \text{ then } S \setminus X \not\in E(N \setminus C)$ (regularity)
It remains to be shown that $E^c(s)$ is playable.

**Proposition: Alternative playability properties**

An effectivity function $E$ is playable iff:

1. $\forall C \neq N: \emptyset \notin E(C)$
2. $\forall C \neq N: S \in E(C)$
3. $\forall X \subseteq S: S \setminus X \notin E(\emptyset) \Rightarrow X \in E(N)$ ($N$-maximality)
4. $\forall C \neq N: \forall X \subseteq X' \subseteq S: X \in E(C) \Rightarrow X' \in E(C)$ (outcome-monotonicity)
5. $\forall C_1 \subset N: \forall C_2 \subset N$ s.t. $C_1 \cup C_2 \neq N: \forall X_1 \subseteq S: \forall X_2 \subseteq S: (C_1 \cap C_2 = \emptyset$ and $X_1 \in E(C_1)$ and $X_2 \in E(C_2)) \Rightarrow X_1 \cap X_2 \in E(C_1 \cup C_2)$ (superadditivity)
6. $\forall C \subseteq N: \forall X \subseteq S: \text{if } X \in E(C) \text{ then } S \setminus X \notin E(N \setminus C)$ (regularity)

Relatively easy to show that these must hold in $\mathcal{M}^c$
Superadditivity

Let \( C_1 \cap C_2 = \emptyset \), \( C_1 \cup C_2 \neq N \), \( X_1 \in E^c(s)(C_1) \), \( X_2 \in E^c(s)(C_2) \):

\[
\langle \langle C_1 \rangle \rangle \varphi_1, \langle \langle C_2 \rangle \rangle \varphi_2 \in s \text{ for some } \varphi_1, \varphi_2 \text{ s.t. } \overline{\varphi}_1 \subseteq X_1 \text{ and } \overline{\varphi}_2 \subseteq X_2
\]

Thus, \( X_1 \cap X_2 \in E^c(s)(C_1 \cup C_2) \).
Superadditivity

Let $C_1 \cap C_2 = \emptyset$, $C_1 \cup C_2 \neq N$, $X_1 \in E^c(s)(C_1)$, $X_2 \in E^c(s)(C_2)$:

- $\langle \langle C_1 \rangle \rangle \phi_1, \langle \langle C_2 \rangle \rangle \phi_2 \in s$ for some $\phi_1, \phi_2$ s.t. $\tilde{\phi}_1 \subseteq X_1$ and $\tilde{\phi}_2 \subseteq X_2$
Superadditivity

Let $C_1 \cap C_2 = \emptyset$, $C_1 \cup C_2 \neq N$, $X_1 \in E^c(s)(C_1)$, $X_2 \in E^c(s)(C_2)$:

- $\langle \langle C_1 \rangle \phi_1, \langle \langle C_2 \rangle \phi_2 \in s \text{ for some } \phi_1, \phi_2 \text{ s.t. } \tilde{\phi}_1 \subseteq X_1 \text{ and } \tilde{\phi}_2 \subseteq X_2$

- By the superadditivity axiom: $\langle \langle C_1 \cup C_2 \rangle \phi_1 \land \phi_2 \rangle \in s$
Superadditivity

Let $C_1 \cap C_2 = \emptyset$, $C_1 \cup C_2 \neq N$, $X_1 \in E^c(s)(C_1)$, $X_2 \in E^c(s)(C_2)$:

- $\langle \langle C_1 \rangle \rangle \phi_1, \langle \langle C_2 \rangle \rangle \phi_2 \in s$ for some $\phi_1, \phi_2$ s.t. $\tilde{\phi}_1 \subseteq X_1$ and $\tilde{\phi}_2 \subseteq X_2$
- By the superadditivity axiom: $\langle \langle C_1 \cup C_2 \rangle \rangle (\phi_1 \land \phi_2) \in s$
- $\phi_1 \land \phi_2 \subseteq X_1 \cap X_2$
Superadditivity

Let $C_1 \cap C_2 = \emptyset$, $C_1 \cup C_2 \neq N$, $X_1 \in E^c(s)(C_1)$, $X_2 \in E^c(s)(C_2)$:

- $\langle \langle C_1 \rangle \rangle \phi_1, \langle \langle C_2 \rangle \rangle \phi_2 \in s$ for some $\phi_1, \phi_2$ s.t. $\tilde{\phi}_1 \subseteq X_1$ and $\tilde{\phi}_2 \subseteq X_2$
- By the superadditivity axiom: $\langle \langle C_1 \cup C_2 \rangle \rangle (\phi_1 \wedge \phi_2) \in s$
- $\phi_1 \wedge \phi_2 \subseteq X_1 \cap X_2$
- Thus, $X_1 \cap X_2 \in E^c(s)(C_1 \cup C_2)$
Computational Complexity

- The problem: 
  
  \textit{Given coalition logic formula }\phi\textit{ is there some model that satisfies }\phi\textit{?}

- Complexity: PSPACE-complete
2.6 Quantified Coalition Logic
Lack of Succinctness in CL

Take the property:

agent 1 is necessary to achieve $\varphi$
Lack of Succinctness in CL

Take the property:

agent 1 is necessary to achieve \( \varphi \)

Its expression in CL is exponentially long in the number of agents in the system. If \( Ag = \{1, 2, 3, 4\} \):

\[
\neg \langle \langle \{\} \rangle \rangle \varphi \land \neg \langle \langle \{2\} \rangle \rangle \varphi \land \neg \langle \langle \{3\} \rangle \rangle \varphi \land \neg \langle \langle \{4\} \rangle \rangle \varphi \land \neg \langle \langle \{2, 3\} \rangle \rangle \varphi \land \\
\neg \langle \langle \{3, 4\} \rangle \rangle \varphi \land \neg \langle \langle \{2, 4\} \rangle \rangle \varphi \land \neg \langle \langle \{2, 3, 4\} \rangle \rangle \varphi
\]
Quantified Coalition Logic (QCL)

In QCL $\langle \cdot \rangle$ is replaced by a collection of unary modal operators indexed by a coalition predicate $P$, in order to make the logic more succinct:

- $\langle P \rangle \varphi$: there exists some coalition satisfying $P$ which can achieve $\varphi$
- $[P] \varphi$: every coalition satisfying $P$ can achieve $\varphi$
Quantified Coalition Logic (QCL)

In QCL $\langle \cdot \rangle$ is replaced by a collection of unary modal operators indexed by a coalition predicate $P$, in order to make the logic more succinct:

- $\langle P \rangle \varphi$: there exists some coalition satisfying $P$ which can achieve $\varphi$
- $[P] \varphi$: every coalition satisfying $P$ can achieve $\varphi$

Examples of predicates ($C'$ a coalition, $n$ a number):

- $\text{supseteq}(C')$: satisfied by $C$ iff $C \supseteq C'$
- $\text{geq}(n)$: satisfied by $C$ iff $|C| \geq n$
- $\text{gt}(n)$: satisfied by $C$ iff $|C| > n$
- $\text{maj}(n) \equiv \text{geq}(\lceil (n + 1)/2 \rceil)$
- Boolean combinations
QCL: Example

agent 1 is necessary to achieve $\varphi$
QCL: Example

agent 1 is necessary to achieve $\varphi$

$\neg \langle \neg \supseteqq \{1\} \rangle \varphi$
QCL Example: voting

An electorate of $n$ voters wishes to select one of two outcomes $\omega_1$ and $\omega_2$. They want to use a simple majority voting protocol, so that outcome $\omega_i$ will be selected iff a majority of the $n$ voters state a preference for it. No coalition of less than majority size should be able to select an outcome, and any majority should be able to choose the outcome (i.e., the selection procedure is not influenced by the “names” of the agents in a coalition).
QCL Example: voting

An electorate of \(n\) voters wishes to select one of two outcomes \(\omega_1\) and \(\omega_2\). They want to use a simple majority voting protocol, so that outcome \(\omega_i\) will be selected iff a majority of the \(n\) voters state a preference for it. No coalition of less than majority size should be able to select an outcome, and any majority should be able to choose the outcome (i.e., the selection procedure is not influenced by the “names” of the agents in a coalition).

\[
([maj(n)]\omega_1) \land ([maj(n)]\omega_2)
\]
QCL Example: voting

An electorate of \( n \) voters wishes to select one of two outcomes \( \omega_1 \) and \( \omega_2 \). They want to use a simple majority voting protocol, so that outcome \( \omega_i \) will be selected iff a majority of the \( n \) voters state a preference for it. No coalition of less than majority size should be able to select an outcome, and any majority should be able to choose the outcome (i.e., the selection procedure is not influenced by the “names” of the agents in a coalition).

\[
([maj(n)]\omega_1) \land ([maj(n)]\omega_2)
\]

\[
(\neg\neg<maj(n)\omega_1) \land (\neg\neg<maj(n)\omega_2)
\]
QCL

- QCL is no more expressive than Coalition Logic
- But is exponentially more succinct
- The model checking problem can be solved in polynomial time – assuming an explicit representation of models
- The model checking problem assuming an RML representation of models is PSPACE-complete.
- The satisfiability problem is PSPACE-complete.
2. Coalition Logic

_A Course in Game Theory_.  

Logic for Social Software.  

A modal logic for coalitional power in games.  

Modal logic for games and information.  

Quantified coalition logic.  
_Synthese_, 2008.
3. ATL

ATL: What Agents Can Achieve

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

- Temporal logic meets game theory
- Main idea: cooperation modalities

\[ \langle A \rangle \Phi: \text{coalition } A \text{ has a collective strategy to enforce } \Phi \]
3.1 The Logic
3. ATL

1. The Logic

Syntax

\[ \varphi ::= p \mid \neg \varphi \mid \varphi \land \varphi \mid \langle A \rangle \gamma, \]
\[ \gamma ::= \varphi \mid \neg \gamma \mid \gamma \land \gamma \mid \Diamond \gamma \mid \Box \gamma \mid \gamma U \gamma. \]
3. ATL

1. The Logic

Syntax

$$\varphi ::= p \mid \neg \varphi \mid \varphi \land \varphi \mid \langle A \rangle \gamma,$$

$$\gamma ::= \varphi \mid \neg \gamma \mid \gamma \land \gamma \mid \Box \gamma \mid \Diamond \gamma \mid \Diamond U \gamma.$$

In fact, “eventually” and “always” can be derived from “until”:

- $$\Diamond \gamma \equiv T U \gamma$$
- $$\Box \gamma \equiv \neg \Diamond \neg \gamma$$
3. ATL

1. The Logic

Syntax

\[ \varphi ::= p \mid \neg \varphi \mid \varphi \land \varphi \mid \langle A \rangle \gamma, \]
\[ \gamma ::= \varphi \mid \neg \gamma \mid \gamma \land \gamma \mid \bigcirc \gamma \mid \diamond \gamma \mid \Box \gamma \mid \gamma U \gamma. \]

In fact, “eventually” and “always” can be derived from “until”:

- \[ \diamond \gamma \equiv \top U \gamma \]
- \[ \Box \gamma \equiv \neg \diamond \neg \gamma \]

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

ATL*: no syntactic restrictions
\( \langle \text{jamesbond} \rangle \Diamond (\text{ski} \land \neg \text{getBurned}) \):

“James Bond can go skiing without getting burned”
\( \langle \text{jamesbond}\rangle \diamond (\text{ski} \land \neg \text{getBurned}) \):  
“James Bond can go skiing without getting burned”
3. ATL

- $\langle j\text{ames}b\text{ond} \rangle \Diamond (\text{ski } \land \neg \text{getBurned})$: “James Bond can go skiing without getting burned”

- $\langle j\text{ames}b\text{ond}, b\text{ondsgi}r\text{rl} \rangle \text{fun } U \text{shot}$: “James Bond and his girlfriend are able to have fun until someone shoots at them”
ATL extends the branching-time logic CTL:

- $A \equiv \langle \emptyset \rangle$ ("for all paths")
- $E \equiv \langle \text{gt} \rangle$ ("there is a path")
3. ATL

ATL Models: Concurrent Game Structures

- Agents, actions, transitions, atomic propositions
- Atomic propositions + interpretation
- Actions are abstract
Definition 3.1 (Concurrent Game Structure)

A concurrent game structure is a tuple
\[ M = \langle \text{Agt}, \text{St}, \pi, \text{Act}, d, o \rangle, \text{ where:} \]

\[ \text{Agt}: \text{a finite set of all agents} \]
\[ \text{St}: \text{a set of states} \]
\[ \pi: \text{a valuation of propositions} \]
\[ \text{Act}: \text{a finite set of (atomic) actions} \]
\[ d: \text{Agt} \times \text{St} \rightarrow 2^{\text{Act}} \text{ defines actions available to an agent in a state} \]
\[ o: \text{a deterministic transition function that assigns outcome states } q' = o(q, \alpha_1, \ldots, \alpha_k) \text{ to states and tuples of actions} \]
Definition 3.1 (Concurrent Game Structure)

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

- $\text{Ag}t$: a finite set of all agents

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

- $\text{Ag}t$: a finite set of all agents
Definition 3.1 (Concurrent Game Structure)

A **concurrent game structure** is a tuple $M = \langle \text{Agt}, St, \pi, \text{Act}, d, o \rangle$, where:

- $\text{Agt}$: a finite set of all agents
- $St$: a set of states
Definition 3.1 (Concurrent Game Structure)

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

- \( \text{Ag}t \): a finite set of all agents
- \( S\text{t} \): a set of states
- \( \pi \): a valuation of propositions

\( d : \text{Ag}t \times S\text{t} \rightarrow 2^{A\text{ct}} \) defines actions available to an agent in a state

\( o : \text{Ag}t \times (S\text{t}, A\text{ct}) \rightarrow (S\text{t}, A\text{ct}) \) assigns outcome states to states and tuples of actions
Definition 3.1 (Concurrent Game Structure)

A concurrent game structure is a tuple $M = \langle \text{Agt}, St, \pi, Act, d, o \rangle$, where:

- $\text{Agt}$: a finite set of all agents
- $St$: a set of states
- $\pi$: a valuation of propositions
- $Act$: a finite set of (atomic) actions
- $d$: $\text{Agt} \times St \rightarrow 2^{Act}$ defines actions available to an agent in a state
- $o$: a deterministic transition function that assigns outcome states $q' = o(q, \alpha_1, ..., \alpha_k)$ to states and tuples of actions
Definition 3.1 (Concurrent Game Structure)  

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

- \( \text{Agt} \): a finite set of all **agents**
- \( St \): a set of **states**
- \( \pi \): a **valuation** of propositions
- \( Act \): a finite set of (atomic) **actions**
- \( d : \text{Agt} \times St \rightarrow 2^{Act} \) 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
Example: Robots and Carriage
Definition 3.2 (Strategy)

A strategy is a conditional plan.
Definition 3.2 (Strategy)

A \textit{strategy} is a \textit{conditional plan}. We represent strategies by functions \( s_a : St \rightarrow Act \).
Definition 3.2 (Strategy)

A strategy is a conditional plan. We represent strategies by functions $s_a : St \rightarrow Act$.

memoryless agents
Definition 3.2 (Strategy)

A strategy is a conditional plan.
We represent strategies by functions $s_a : St \rightarrow Act$.

$\leadsto$ memoryless agents

Alternative: perfect recall strategies $s_a : St^+ \rightarrow Act$
Definition 3.2 (Strategy)

A **strategy** is a conditional plan. We represent strategies by functions $s_a : St \rightarrow Act$.

Alternative: **perfect recall strategies** $s_a : St^+ \rightarrow Act$

Function $out(q, s_A)$ returns the set of all paths that may result from agents $A$ executing strategy $s_A$ from state $q$ onward.
Definition 3.3 (Semantics of ATL*)

\[ M, q \models p \iff p \text{ is in } \pi(q); \]
\[ M, q \models \neg \varphi \iff M, q \not\models \varphi; \]
\[ M, q \models \varphi_1 \land \varphi_2 \iff M, q \models \varphi_1 \text{ and } M, q \models \varphi_2; \]
Definition 3.3 (Semantics of ATL*)

\[ M, q \models p \iff p \text{ is in } \pi(q); \]
\[ M, q \models \neg \varphi \iff M, q \not\models \varphi; \]
\[ M, q \models \varphi_1 \land \varphi_2 \iff M, q \models \varphi_1 \text{ and } M, q \models \varphi_2; \]
\[ M, \lambda \models \neg \gamma \iff M, q \not\models \gamma \text{ etc.}; \]
Definition 3.3 (Semantics of ATL*)

\[ M, q \models p \ \text{iff} \ p \text{ is in } \pi(q); \]
\[ M, q \models \neg \varphi \ \text{iff} \ M, q \not\models \varphi; \]
\[ M, q \models \varphi_1 \land \varphi_2 \ \text{iff} \ M, q \models \varphi_1 \text{ and } M, q \models \varphi_2; \]
\[ M, \lambda \models \neg \gamma \ \text{iff} \ M, q \not\models \gamma \text{ etc.}; \]
\[ M, q \models \langle A \rangle \Phi \ \text{iff there is a collective strategy } s_A \text{ such that, for every path } \lambda \in \text{out}(q, s_A), \text{ we have } M, \lambda \models \Phi. \]
Definition 3.3 (Semantics of ATL*)

\( M, q \models p \) \iff \( p \) is in \( \pi(q) \);

\( M, q \models \neg \varphi \) \iff \( M, q \not\models \varphi \);

\( M, q \models \varphi_1 \land \varphi_2 \) \iff \( M, q \models \varphi_1 \) and \( M, q \models \varphi_2 \);

\( M, \lambda \models \neg \gamma \) \iff \( M, q \not\models \gamma \) etc.;

\( M, q \models \langle A \rangle \Phi \) \iff there is a collective strategy \( s_A \) such that, for every path \( \lambda \in \text{out}(q, s_A) \), we have \( M, \lambda \models \Phi \).

\( M, \lambda \models \Box \gamma \) \iff \( M, \lambda[1..\infty] \models \gamma \);
Definition 3.3 (Semantics of ATL*)

\( M, q \models p \) iff \( p \) is in \( \pi(q) \);

\( M, q \models \neg \varphi \) iff \( M, q \not\models \varphi \);

\( M, q \models \varphi_1 \land \varphi_2 \) iff \( M, q \models \varphi_1 \) and \( M, q \models \varphi_2 \);

\( M, \lambda \models \neg \gamma \) iff \( M, q \not\models \gamma \) etc.;

\( M, q \models \unlhd A \Phi \) iff there is a collective strategy \( s_A \) such that, for every path \( \lambda \in \text{out}(q, s_A) \), we have \( M, \lambda \models \Phi \).

\( M, \lambda \models \Diamond \gamma \) iff \( M, \lambda[1..\infty] \models \gamma \);

\( M, \lambda \models \Box \gamma \) iff \( M, \lambda[i..\infty] \models \gamma \) for all \( i \geq 0 \);
Definition 3.3 (Semantics of ATL*)

\[ M, q \models p \iff p \text{ is in } \pi(q); \]
\[ M, q \models \neg \varphi \iff M, q \not\models \varphi; \]
\[ M, q \models \varphi_1 \land \varphi_2 \iff M, q \models \varphi_1 \text{ and } M, q \models \varphi_2; \]
\[ M, \lambda \models \neg \gamma \iff M, q \not\models \gamma \text{ etc.}; \]
\[ M, q \models \langle A \rangle \Phi \iff \text{there is a collective strategy } s_A \text{ such that, for every path } \lambda \in \text{out}(q, s_A), \text{ we have } M, \lambda \models \Phi. \]
\[ M, \lambda \models \bigcirc \gamma \iff M, \lambda[1..\infty] \models \gamma; \]
\[ M, \lambda \models \Box \gamma \iff M, \lambda[i..\infty] \models \gamma \text{ for all } i \geq 0; \]
\[ M, \lambda \models \gamma_1 \mathcal{U} \gamma_2 \iff M, \lambda[i..\infty] \models \gamma_2 \text{ for some } i \geq 0, \text{ and } M, \lambda[j..\infty] \models \gamma_1 \text{ for all } 0 \leq j \leq i. \]
The semantics of “vanilla” ATL can be given entirely in terms of models and states:

\[ M, q \models p \iff p \text{ is in } \pi(q); \]
\[ M, q \models \neg \varphi \iff M, q \not\models \varphi; \]
\[ M, q \models \varphi_1 \land \varphi_2 \iff M, q \models \varphi_1 \text{ and } M, q \models \varphi_2; \]
\[ M, q \models \langle \langle A \rangle \rangle \Box \varphi \iff \text{there is } s_A \text{ such that, for every } \lambda \in \text{out}(q, s_A), \text{ we have } M, \lambda[1] \models \varphi; \]
\[ M, q \models \langle \langle A \rangle \rangle \varphi \iff \text{there is } s_A \text{ such that, for every } \lambda \in \text{out}(q, s_A), \text{ we have } M, \lambda[i] \models \varphi \text{ for all } i \geq 0; \]
\[ M, q \models \langle \langle A \rangle \rangle \varphi_1 U \varphi_2 \iff \text{there is } s_A \text{ such that, for every } \lambda \in \text{out}(q, s_A), \text{ we have } M, \lambda[i] \models \varphi_2 \text{ for some } i \geq 0 \text{ and } M, \lambda[j] \models \varphi_1 \text{ for all } 0 \leq j \leq i. \]
Example: Robots and Carriage

\[ \text{pos}_0 \rightarrow \langle 1 \rangle \square \neg \text{pos}_1 \]
Example: Robots and Carriage

\[ \text{pos}_0 \rightarrow \langle 1 \rangle \square \neg \text{pos}_1 \]
Example: Robots and Carriage

\[ \text{pos}_0 \rightarrow \langle 1 \rangle \Box \neg \text{pos}_1 \]
Example: Robots and Carriage

\[ \text{pos}_0 \rightarrow \langle 1 \rangle \square \neg \text{pos}_1 \]
Example: Robots and Carriage

\[ \text{pos}_0 \rightarrow \langle 1 \rangle \Box \neg \text{pos}_1 \]
Example: Robots and Carriage

\[ \text{pos}_0 \rightarrow \langle 1 \rangle \Box \neg \text{pos}_1 \]
Example: Robots and Carriage

\[ \text{pos}_0 \rightarrow \langle 1 \rangle \Box \neg \text{pos}_1 \]
Fixpoint Properties

**Theorem 3.4**

The following formulae are valid for ATL (but not for ATL*!):

- $\langle \langle A \rangle \rangle \Box \varphi \iff \varphi \land \langle \langle A \rangle \rangle \bigotimes \langle \langle A \rangle \rangle \Box \varphi$
- $\langle \langle A \rangle \rangle \varphi_1 U \varphi_2 \iff \varphi_2 \lor \varphi_1 \land \langle \langle A \rangle \rangle \bigotimes \langle \langle A \rangle \rangle \varphi_1 U \varphi_2$. 
3. ATL

Fixpoint Properties

**Theorem 3.4**

The following formulae are valid for ATL (but not for ATL*!):

- $\langle \langle A \rangle \rangle \Box \phi \leftrightarrow \phi \land \langle \langle A \rangle \rangle \bigcirc \langle \langle A \rangle \rangle \Box \phi$
- $\langle \langle A \rangle \rangle \phi_1 U \phi_2 \leftrightarrow \phi_2 \lor \phi_1 \land \langle \langle A \rangle \rangle \bigcirc \langle \langle A \rangle \rangle \phi_1 U \phi_2$.

**Corollary**

Strategy for $A$ can be synthesized incrementally (no backtracking is necessary).
3.2 Agents, Systems, Games
3. ATL

2. Agents, Systems, Games

Connection to Temporal Analysis of Systems

Temporal operators allow a number of useful concepts to be formally specified:
Connection to Temporal Analysis of Systems

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”
Safety (maintenance goals):

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

Typical example:

□¬bankrupt
Safety (maintenance goals):

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

Typical example:

□¬bankrupt

Usually: □¬....
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”
Liveness (achievement goals):

“something good will happen”

Typical example:

◊ rich

Usually: ◊ ....
Liveness (achievement goals):
“something good will happen”

Typical example:
◊ rich

Usually: ◊ ....

In ATL:

⟨ ⟨alice, bob⟩⟩◊paperAccepted
Fairness (service goals):

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

Fairness (service goals):

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

Typical examples:

\( \Box (\text{attempt} \rightarrow \Diamond \text{success}) \)
\( \Box \Diamond \text{attempt} \rightarrow \Box \Diamond \text{success} \)
Fairness (service goals):

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

Typical examples:

\[ \square (\text{attempt} \rightarrow \Diamond \text{success}) \]
\[ \square \Diamond \text{attempt} \rightarrow \square \Diamond \text{success} \]

In ATL* (!):

\[ \langle \langle \text{prod}, \text{dlr} \rangle \rangle \square (\text{carRequested} \rightarrow \Diamond \text{carDelivered}) \]
Connection to Multi-Agent/Multi-Process Systems

- Validity $\iff$ General properties of systems
Connection to Multi-Agent/Multi-Process Systems

- **Validity** $\iff$ General properties of systems
- **Satisfiability** $\iff$ System synthesis
Connection to Multi-Agent/Multi-Process Systems

- Validity $\iff$ General properties of systems
- Satisfiability $\iff$ System synthesis
- Model checking $\iff$ Verification
Connection to Multi-Agent/Multi-Process Systems

- Validity ⇔ General properties of systems
- Satisfiability ⇔ System synthesis
- Model checking ⇔ Verification

ATL is just another specification language in this context...
Connection to Games

- Concurrent game structure = generalized extensive game
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
Connection to Games

- Concurrent game structure = generalized extensive game
- $\langle A \rangle \gamma$: $\langle A \rangle$ splits the agents into proponents and opponents
- $\gamma$ defines the winning condition
  $\rightsquigarrow$ infinite 2-player, binary, zero-sum game
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
  \( \rightsquigarrow \) infinite 2-player, binary, zero-sum game
- Flexible and compact specification of winning conditions
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 $M, q \models \langle \langle A \rangle \rangle \gamma$
- Model checking ATL corresponds to game solving in game theory!
Connection to Games

What about other problems?
Connection to Games

What about other problems?

- Validity $\iff$ General properties of games
Connection to Games

What about other problems?

- **Validity** $\iff$ General properties of games
- **Satisfiability** $\iff$ Mechanism design
Connection to Games

What about other problems?

- **Validity** ⇔ **General properties of games**
- **Satisfiability** ⇔ **Mechanism design**
- **E.g., building a model for** $\langle \emptyset \rangle \gamma_1 \land \langle A \rangle \gamma_1$ **Designing a game in which** $\gamma_1$ **is guaranteed and** $A$ **can achieve** $\gamma_2$
3.3 A Short Look at Satisfiability
Satisfiability of Temporal and Strategic Logics: Complexity Results

<table>
<thead>
<tr>
<th>Logic</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTL</td>
<td>EXPTIME-complete</td>
</tr>
<tr>
<td>LTL</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>CTL*</td>
<td>2EXPTIME-complete</td>
</tr>
<tr>
<td>ATL</td>
<td>EXPTIME-complete</td>
</tr>
<tr>
<td>ATL*</td>
<td>2EXPTIME-complete</td>
</tr>
</tbody>
</table>
### Satisfiability of Temporal and Strategic Logics: Complexity Results

<table>
<thead>
<tr>
<th>Logic</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTL</td>
<td>EXPTIME-complete</td>
</tr>
<tr>
<td>LTL</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>CTL*</td>
<td>2EXPTIME-complete</td>
</tr>
<tr>
<td>ATL</td>
<td>EXPTIME-complete</td>
</tr>
<tr>
<td>ATL*</td>
<td>?</td>
</tr>
</tbody>
</table>

For strategies with perfect recall:

- ATL  EXPTIME-complete
- ATL* 2EXPTIME-complete
Satisfiability of Temporal and Strategic Logics: Complexity Results

<table>
<thead>
<tr>
<th></th>
<th>( l )</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTL</td>
<td>EXPTIME-complete</td>
</tr>
<tr>
<td>LTL</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>CTL*</td>
<td>2EXPTIME-complete</td>
</tr>
<tr>
<td>ATL</td>
<td>EXPTIME-complete</td>
</tr>
<tr>
<td>ATL*</td>
<td>?</td>
</tr>
</tbody>
</table>

For strategies with perfect recall:

<table>
<thead>
<tr>
<th></th>
<th>( m, l )</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATL</td>
<td>EXPTIME-complete</td>
</tr>
<tr>
<td>ATL*</td>
<td>2EXPTIME-complete</td>
</tr>
</tbody>
</table>
For the Interested Ones...


3.4 References

More about ATL
4. More about ATL

1. Axiomatisation

4.1 Axiomatisation
4. More about ATL

1. Axiomatisation

Sound and Compl. Ax. (Goranko, van Drimmelen)

\[\begin{align*}
(\bot) & \quad \neg \langle C \rangle \circ \bot \\
(\top) & \quad \langle C \rangle \circ \top \\
(\Sigma) & \quad \neg \langle \emptyset \rangle \circ \neg \varphi \rightarrow \langle \Sigma \rangle \circ \varphi \\
(S) & \quad \langle C_1 \rangle \circ \varphi_1 \land \langle C_2 \rangle \circ \varphi_2 \rightarrow \langle C_1 \cup C_2 \rangle \circ (\varphi_1 \land \varphi_2), \\
\end{align*}\]

where \(C_1\) and \(C_2\) are disjoint

\[\begin{align*}
\varphi_1, \varphi_1 \rightarrow \varphi_2 \quad (MP) & \quad \varphi_1 \rightarrow \varphi_2 \quad (Mon) \\
\varphi_2 \quad & \quad \langle C \rangle \circ \varphi_1 \rightarrow \langle C \rangle \circ \varphi_2 \\
\end{align*}\]
4. More about ATL

Sound and Compl. Ax. (Goranko, van Drimmelen)

\[
\begin{align*}
(\bot) & \quad \neg \langle\langle C'\rangle\rangle \circ \bot \\
(\top) & \quad \langle\langle C'\rangle\rangle \circ \top \\
(\Sigma) & \quad \neg \langle\langle \emptyset \rangle\rangle \circ \neg \varphi \rightarrow \langle\langle \Sigma \rangle\rangle \circ \varphi \\
(S) & \quad \langle\langle C_1\rangle\rangle \circ \varphi_1 \land \langle\langle C_2\rangle\rangle \circ \varphi_2 \rightarrow \langle\langle C_1 \cup C_2\rangle\rangle \circ (\varphi_1 \land \varphi_2), \\
& \quad \text{where } C_1 \text{ and } C_2 \text{ are disjoint} \\
(FP\Box) & \quad \langle\langle C'\rangle\rangle \Box \varphi \leftrightarrow \varphi \land \langle\langle C'\rangle\rangle \circ \langle\langle C'\rangle\rangle \Box \varphi
\end{align*}
\]

\[
\begin{align*}
\varphi_1, \varphi_1 \rightarrow \varphi_2 & \quad (MP) \\
\varphi_1 \rightarrow \varphi_2 & \quad (Mon) \\
\langle\langle C'\rangle\rangle \circ \varphi_1 \rightarrow \langle\langle C'\rangle\rangle \circ \varphi_2 & \\
\langle\langle \emptyset \rangle\rangle \Box \varphi & \quad (Nec)
\end{align*}
\]
4. More about ATL

1. Axiomatisation

Sound and Compl. Ax. (Goranko, van Drimmelen)

\((\bot)\) \(\neg \langle \langle C \rangle \rangle \circ \bot\)

\((\top)\) \(\langle \langle C \rangle \rangle \circ \top\)

\((\Sigma)\) \(\neg \langle \langle \emptyset \rangle \rangle \circ \neg \varphi \rightarrow \langle \langle \Sigma \rangle \rangle \circ \varphi\)

\((S)\) \(\langle \langle C_1 \rangle \rangle \circ \varphi_1 \land \langle \langle C_2 \rangle \rangle \circ \varphi_2 \rightarrow \langle \langle C_1 \cup C_2 \rangle \rangle \circ (\varphi_1 \land \varphi_2)\),

where \(C_1\) and \(C_2\) are disjoint

\((FP\Box)\) \(\langle \langle C' \rangle \rangle \Box \varphi \leftrightarrow \varphi \land \langle \langle C' \rangle \rangle \circ \langle \langle C' \rangle \rangle \Box \varphi\)

\((GFP\Box)\) \(\langle \langle \emptyset \rangle \rangle \Box (\theta \rightarrow (\varphi \land \langle \langle C' \rangle \rangle \circ \theta)) \rightarrow \langle \langle \emptyset \rangle \rangle \Box (\theta \rightarrow \langle \langle C' \rangle \rangle \Box \varphi)\)

\(\varphi_1, \varphi_1 \rightarrow \varphi_2\) \((MP)\)

\(\varphi_1 \rightarrow \varphi_2\) \((Mon)\)

\(\varphi\) \((Nec)\)
4. More about ATL

1. Axiomatisation

Sound and Compl. Ax. (Goranko, van Drimmelen)

(⊥) \( \neg \langle \langle C \rangle \rangle \circ \downarrow \)

(⊤) \( \langle \langle C \rangle \rangle \circ \top \)

(Σ) \( \neg \langle \langle \emptyset \rangle \rangle \circ \neg \varphi \rightarrow \langle \langle \Sigma \rangle \rangle \circ \varphi \)

(S) \( \langle \langle C_1 \rangle \rangle \circ \varphi_1 \land \langle \langle C_2 \rangle \rangle \circ \varphi_2 \rightarrow \langle \langle C_1 \cup C_2 \rangle \rangle \circ (\varphi_1 \land \varphi_2) \),
where \( C_1 \) and \( C_2 \) are disjoint

(FP□) \( \langle \langle C \rangle \rangle \Box \varphi \leftrightarrow \varphi \land \langle \langle C \rangle \rangle \circ \langle \langle C \rangle \rangle \Box \varphi \)

(GFP□) \( \langle \langle \emptyset \rangle \rangle \Box (\theta \rightarrow (\varphi \land \langle \langle C \rangle \rangle \circ \theta)) \rightarrow \langle \langle \emptyset \rangle \rangle \Box (\theta \rightarrow \langle \langle C \rangle \rangle \Box \varphi) \)

(FPU) \( \langle \langle C \rangle \rangle (\varphi_1 \mathbin{U} \varphi_2) \leftrightarrow \varphi_2 \lor (\varphi_1 \land \langle \langle C \rangle \rangle \circ \langle \langle C \rangle \rangle (\varphi_1 \mathbin{U} \varphi_2)) \)

\[
\begin{align*}
\varphi_1, \varphi_1 & \rightarrow \varphi_2 \quad (MP) \\
\varphi_1 & \rightarrow \varphi_2 \quad (Mon) \\
\langle \langle C \rangle \rangle \circ \varphi_1 & \rightarrow \langle \langle C \rangle \rangle \circ \varphi_2 \quad (Mon) \\
\varphi & \Box \varphi \quad (Nec)
\end{align*}
\]
4. More about ATL

1. Axiomatisation

Sound and Compl. Ax. (Goranko, van Drimmelen)

\[(\bot) \quad \neg \langle \langle C \rangle \rangle \circ \bot\]

\[(\top) \quad \langle \langle C \rangle \rangle \circ \top\]

\[(\Sigma) \quad \neg \langle \langle \emptyset \rangle \rangle \circ \neg \varphi \rightarrow \langle \langle \Sigma \rangle \rangle \circ \varphi\]

\[(S) \quad \langle \langle C_1 \rangle \rangle \circ \varphi_1 \land \langle \langle C_2 \rangle \rangle \circ \varphi_2 \rightarrow \langle \langle C_1 \cup C_2 \rangle \rangle \circ (\varphi_1 \land \varphi_2),\]
where \(C_1\) and \(C_2\) are disjoint

\[(FP_\Box) \quad \langle \langle C' \rangle \rangle \Box \varphi \leftrightarrow \varphi \land \langle \langle C' \rangle \rangle \circ \langle \langle C' \rangle \rangle \Box \varphi\]

\[(GFP_\Box) \quad \langle \langle \emptyset \rangle \rangle \Box (\theta \rightarrow (\varphi \land \langle \langle C' \rangle \rangle \circ \theta)) \rightarrow \langle \langle \emptyset \rangle \rangle \Box (\theta \rightarrow \langle \langle C \rangle \rangle \Box \varphi)\]

\[(FP_\cup) \quad \langle \langle C' \rangle \rangle (\varphi_1 \cup \varphi_2) \leftrightarrow \varphi_2 \lor (\varphi_1 \land \langle \langle C' \rangle \rangle \circ \langle \langle C \rangle \rangle (\varphi_1 \cup \varphi_2))\]

\[(LFP_\cup) \quad \langle \langle \emptyset \rangle \rangle \Box ((\varphi_2 \lor (\varphi_1 \land \langle \langle C' \rangle \rangle \circ \theta)) \rightarrow \theta) \rightarrow \langle \langle \emptyset \rangle \rangle \Box (\langle \langle C \rangle \rangle (\varphi_1 \cup \varphi_2) \rightarrow \theta)\]

\[\frac{\varphi_1, \varphi_1 \rightarrow \varphi_2}{\varphi_2} (MP) \quad \frac{\varphi_1 \rightarrow \varphi_2}{\langle \langle C' \rangle \rangle \circ \varphi_1 \rightarrow \langle \langle C' \rangle \rangle \circ \varphi_2} (Mon) \quad \frac{\varphi}{\langle \langle \emptyset \rangle \rangle \Box \varphi} (Nec)\]
4. More about ATL

2. Bisimulation and The Role of Memory

4.2 Bisimulation and The Role of Memory
Definitions

When $\bar{a}_C \in D(q, C)$ let

$$\text{next}_M(q, \bar{a}_C) = \{ \delta(q, \bar{b}) : \bar{b} \in D(q), a_i = b_i \text{ for all } i \in C \}$$

denote the set of possible next states in CGS $M$ when coalition $C$ choose actions $\bar{a}_C$. 
Definition 4.1 (Bisimulation)

Given CGS $M_1 = (Q_1, \pi_1, act_1, d_1, \delta_1)$; CGS $M_2 = (Q_2, \pi_2, act_2, d_2, \delta_2)$; $\beta \subseteq Q_1 \times Q_2$.

$M_1 \leftrightarrow^C \beta M_2$ (for some $C \subseteq \Sigma$): for all $q_1, q_2$, $q_1 \beta q_2$ implies that

- **Local harmony** $\pi_1(q_1) = \pi_2(q_2)$;
- **Forth** For all joint actions $\vec{a}_C^1 \in D_1(q_1, C)$ for $C$, there exists a joint action $\vec{a}_C^2 \in D_2(q_2, C)$ for $C$ such that for all states $s_2 \in \text{next}_M(q_2, \vec{a}_C^2)$, there exists a state $s_1 \in \text{next}_M(q_1, \vec{a}_C^1)$ such that $s_1 \beta s_2$;
- **Back** Likewise, for 1 and 2 swapped.

$M_1 \leftrightarrow \beta M_2$: $M_1 \leftrightarrow^C \beta M_2$ for every $C \subseteq \Sigma$
Bisimulation: Example

\[ \beta = \{(q_1, q'_1), (q_2, q'_2), (q_4, q'_2), (q_3, q'_3)\} \]
Strategies and Memory

Let us discern between two definitions of the satisfaction relation:

\( \models^F: \) perfect recall is assumed, all strategies

\[ f : Q^+ \rightarrow \text{act} \]

are allowed

\( \models^L: \) only memoryless strategies are allowed, i.e., strategies

\[ f : Q \rightarrow \text{act} \]
Invariance under Bisimulation: the Memoryless Case

**Theorem 4.2 (Bisimulation Characterisation)**

If $M_1 \mathrel{\beta} M_2$ and $s_1 \mathrel{\beta} s_2$, then for every ATL formula $\varphi$:

$$M_1, s_1 \models_L \varphi \iff M_2, s_2 \models_L \varphi$$
4. More about ATL

2. Bisimulation and The Role of Memory

Tree-unfolding

Let $\text{fincomp}_M(q)$ denote the set of finite prefixes of computations starting in $q$. Let $\ell(q_0 \cdots q_k) = q_k$.

Definition 4.3 (Tree-unfolding of CGS)

Given a CGS

\[ M = (Q, \pi, \text{act}, d, \delta) \]

and $q \in Q$, the tree-unfolding $T(M, q)$ of $M$ from $q$ is defined as follows:

\[ T(M, q) = (Q^*, \pi^*, \text{act}, d^*, \delta^*) \]

where $Q^* = \text{fincomp}_M(q)$; $\pi^*(\sigma) = \pi(\ell(\sigma))$; $d^*_i(\sigma) = d_i(\ell(\sigma))$; and $\delta^*(\sigma, a) = \sigma \delta(\ell(\sigma), a)$. 
Lemma 4.4

For any $\mathcal{M}$, $q$,

$$T(\mathcal{M}, q) \leftrightarrow_\beta \mathcal{M}$$

where $\beta = \{(\sigma, \ell(\sigma)) \mid \sigma \in fincomp_{\mathcal{M}}(q)\}$
Lemma 4.5

For any \( \mathcal{M}, q \) and \( \varphi \),

\[
T(\mathcal{M}, q), q \models_L \varphi \iff \mathcal{M}, q \models_F \varphi
\]
Memory Does not Influence Ability

**Corollary 4.6**

For any $\mathcal{M}$, $q$ and $\varphi$,

$$\mathcal{M}, q \models_L \varphi \iff \mathcal{M}, q \models_F \varphi$$

Also: the axiomatisation is sound and complete wrt. both semantics.
Memory Does not Influence Ability

**Corollary 4.6**

For any $\mathcal{M}, q$ and $\varphi$,

$$\mathcal{M}, q \models L \varphi \iff \mathcal{M}, q \models F \varphi$$

Also: the axiomatisation is sound and complete wrt. both semantics.
Invariance under Bisimulation: the Perfect Recall Case

**Corollary 4.7**

If $M_1 \equiv^\beta M_2$ and $s_1 \beta s_2$, then

$$M_1, s_1 \models F \varphi \iff M_2, s_2 \models F \varphi$$

for every ATL formula $\varphi$. 
ATL* and memory

For ATL* – contrary to ATL – memory matters:
4. More about ATL

2. Bisimulation and The Role of Memory

ATL* and memory

For ATL* – contrary to ATL – memory matters:

Proposition

There is a model $M$ with a state $q$, and a formula $\varphi$, such that

$$M, q \models L \varphi \iff M, q \models F \varphi$$
ATL* and memory

For ATL* — contrary to ATL — memory matters:

Proposition

There is a model $M$ with a state $q$, and a formula $\varphi$, such that

$M, q \models_L \varphi \not\equiv M, q \models_F \varphi$

$\varphi = \langle a \rangle (\Box p \land \Diamond \Diamond \neg p)$
4.3 Revocability of Strategies
Example

- $p$: agent $a$ controls the resource
- $\langle a \rangle \bigcirc p$: $a$ has the ability to control the resource next
- $\langle a \rangle \Box \langle a \rangle \bigcirc p$: $a$ has the ability to ensure that $\langle a \rangle \bigcirc p$ is always true
4. More about ATL

3. Revocability of Strategies

Example

- \(p\): agent \(a\) controls the resource
- \(\langle a \rangle \bigcirc p\): \(a\) has the ability to control the resource next
- \(\langle a \rangle \Box \langle a \rangle \bigcirc p\): \(a\) has the ability to ensure that \(\langle a \rangle \bigcirc p\) is always true

\[ M : \]

- \(q_1\): \(\neg p\)
- \(q_2\): \(p\)
- \(q_3\): \(\neg p\)

Arrows:
- \(\alpha_1\) from \(q_1\) to \(q_3\)
- \(\alpha_2\) from \(q_1\) to \(q_2\)
- \(\alpha_1\) from \(q_2\) to \(q_3\)

Counterintuitive?

\(a\) can ensure that she is forever able to access the resource – but only without never actually accessing it.
Example

- \( q \): agent \( a \) controls the resource
- \( \langle a \rangle \Box p \): \( a \) has the ability to control the resource next
- \( \langle a \rangle \Box \langle a \rangle \Box p \): \( a \) has the ability to ensure that \( \langle a \rangle \Box p \) is always true

\[ M, q_1 \models \langle a \rangle \Box p \]
4. More about ATL

Example

- \( p \): agent \( a \) controls the resource
- \( \langle \langle a \rangle \rangle \lozenge p \): \( a \) has the ability to control the resource next
- \( \langle \langle a \rangle \rangle \Box \langle \langle a \rangle \rangle \lozenge p \): \( a \) has the ability to ensure that \( \langle \langle a \rangle \rangle \lozenge p \) is always true

\[ M, q_1 \models \langle \langle a \rangle \rangle \lozenge p \]
\[ M, q_1 \models \langle \langle a \rangle \rangle \Box \langle \langle a \rangle \rangle \lozenge p \]
4. More about ATL

Example

- **p**: agent $a$ controls the resource
- **$\langle \langle a \rangle \rangle \bigcirc p$**: $a$ has the ability to control the resource next
- **$\langle \langle a \rangle \rangle \Box \langle \langle a \rangle \rangle \bigcirc p$**: $a$ has the ability to ensure that $\langle \langle a \rangle \rangle \bigcirc p$ is always true

Counterintuitive? $a$ can ensure that she is forever able to access the resource – but only without never actually accessing it.
In the evaluation of a formula such as $\langle a \rangle \Box \varphi$, when the goal $\varphi$ is evaluated the agent ($a$) is no longer restricted by the strategy she chose in order to get to the state where the goal is evaluated (as the example illustrates).

In this sense, strategies in ATL are revocable.

In some contexts, it would be more natural to reason about strategies which are not revocable and completely specify the future behaviour of the agent.
Alternative: Irrevocable Strategies

Irrevocable strategies can be modelled by using model updates in the semantics.
Alternative: Irrevocable Strategies

Irrevocable strategies can be modelled by using **model updates** in the semantics. Assume memoryless strategies (for now).

**Definition 4.8 (Model Update)**

Let $M$ be a CGS, $C$ a coalition, and $f_C$ a memoryless strategy for $C$. The update of $M$ by $f_C$, denoted $M \uparrow f_C$, is the same as $M$, except that the choices of each agent $i \in C$ are fixed by the strategy $f_i$:

$$d_i(q) = \{ f_i(q) \}$$

for each state $q$. 
Model Update: Example

\[ \mathcal{M} : \begin{align*}
q_1 & \xrightarrow{\neg p} q_2 \\
q_2 & \xrightarrow{p} q_3 \\
q_3 & \xrightarrow{\neg p}
\end{align*} \]

\[ f_1 : \{ q_1 \mapsto \alpha_1, q_2 \mapsto \alpha_1, q_3 \mapsto \alpha_1 \} \]
Model Update: Example

\[ f_1 = \{ q_1 \mapsto \alpha_1, q_2 \mapsto \alpha_1, q_3 \mapsto \alpha_1 \} \]
Model Update: Example

\[ M : \]

\[ q_1 \xrightarrow{\neg p} q_2 \xrightarrow{p} q_3 \xrightarrow{\neg p} \]

\[ \alpha_1 \quad \alpha_2 \quad \alpha_1 \]

\[ f_1 = \{ q_1 \leftrightarrow \alpha_1, q_2 \leftrightarrow \alpha_1, q_3 \leftrightarrow \alpha_1 \} \]

\[ M \upharpoonright f_1 : \]

\[ q_1 \xrightarrow{\neg p} \]

\[ \alpha_1 \]
Satisfiability under Irrevocable Strategies

We can now define a new variant of the satisfiability relation:

\[ \mathcal{M}, q \models_i \langle \langle C \rangle \rangle \Box \phi \iff \exists f_C \forall \lambda \in \text{comp}(\mathcal{M} \uparrow f_C, q, f_C) \]
\[ (\mathcal{M} \uparrow f_C, \lambda[1] \models_i \phi) \]

\[ \mathcal{M}, q \models_i \langle \langle C' \rangle \rangle \Box \phi \iff \exists f_C \forall \lambda \in \text{comp}(\mathcal{M} \uparrow f_C, q, f_C) \]
\[ \forall j \geq 0(\mathcal{M} \uparrow f_C, \lambda[j] \models_i \phi) \]

\[ \mathcal{M}, q \models_i \langle \langle C \rangle \rangle (\phi_1 U \phi_2) \iff \exists f_C \forall \lambda \in \text{comp}(\mathcal{M} \uparrow f_C, q, f_C) \]
\[ \exists j \geq 0(\mathcal{M} \uparrow f_C, \lambda[j] \models_i \phi_2 \text{ and } \]
\[ \forall 0 \leq k < j (\mathcal{M} \uparrow f_C, \lambda[k] \models_i \phi_1) \]
Example (contd.)

\[ M : \]

\[ q_1 \xrightarrow{\lnot p} q_2 \xrightarrow{p} q_3 \xrightarrow{\lnot p} \]

\[ M \]

\[ f_1 = \{ q_1 \mapsto \alpha_1, q_2 \mapsto \alpha_1, q_3 \mapsto \alpha_1 \} \]

\[ f_2 = \{ q_1 \mapsto \alpha_2, q_2 \mapsto \alpha_1, q_3 \mapsto \alpha_1 \} \]
Example (contd.)

\[ M : q_1 \xrightarrow{\alpha_2} q_2 \xrightarrow{\alpha_1} q_3 \]

\[ f_1 = \{ q_1 \leftrightarrow \alpha_1, q_2 \leftrightarrow \alpha_1, q_3 \leftrightarrow \alpha_1 \} \]

\[ f_2 = \{ q_1 \leftrightarrow \alpha_2, q_2 \leftrightarrow \alpha_1, q_3 \leftrightarrow \alpha_1 \} \]
Example (contd.)

\[ M : \]

\[ \begin{align*}
    \alpha_1 & \quad q_1 \\
    \neg p & \quad \alpha_2 \\
    q_2 & \quad p \\
    \alpha_1 & \quad q_3 \\
    \neg p & \\
\end{align*} \]

\[ f_1 = \{ q_1 \mapsto \alpha_1, q_2 \mapsto \alpha_1, q_3 \mapsto \alpha_1 \} \]

\[ f_2 = \{ q_1 \mapsto \alpha_2, q_2 \mapsto \alpha_1, q_3 \mapsto \alpha_1 \} \]

\[ M \upharpoonright f_1 : \]

\[ M \upharpoonright f_2 : \]

\[ \begin{align*}
    \alpha_1 & \quad q_1 \\
    \neg p & \quad q_2 \\
    p & \quad q_3 \\
\end{align*} \]

\[ M, q_1 \models \langle a \rangle \square \langle a \rangle \mathbf{O} p \quad \text{(standard definition)} \]

\[ M, q_1 \not\models_i \langle a \rangle \square \langle a \rangle \mathbf{O} p \quad \text{(with irrevocable strategies)} \]
With irrevocable strategies, truth of formulae is not invariant under bisimulations:

\[ M : q_1 \not\alpha, \beta \]

\[ q_2 \]

\[ q_3 \not\alpha_1, \beta \]

\[ q_4 \alpha, \beta \]

\[ q_5 \alpha, \beta_2 \]

\[ q_2' \alpha_1, \beta_1 \]

\[ q_3' \not\alpha_1, \beta \]

\[ q_4' \alpha, \beta_1 \]

\[ q_5' \alpha, \beta_2 \]

\[ M' : q_1' \not\alpha, \beta \]

\[ q_2' \alpha_1, \beta \]

\[ q_3' \not\alpha_1, \beta_1 \]

\[ q_4' \alpha, \beta \]

\[ q_5' \not\alpha_1, \beta_2 \]

\[ M, q_1 \models_i \langle 1 \rangle \bigcirc (\langle 2 \rangle \bigcirc \langle 0 \rangle \bigcirc \langle 0 \rangle \bigcirc \not p) \land \langle 2 \rangle \bigcirc \langle 0 \rangle \bigcirc \langle 0 \rangle \bigcirc p \]

(strategies: \( \{ q_3 \leftrightarrow \alpha_1, q_5 \leftrightarrow \alpha_2 \}; \{ q_2 \leftrightarrow \beta_1 \}; \{ q_2 \leftrightarrow \beta_2 \} \))
With irrevocable strategies, truth of formulae is not invariant under bisimulations:

\[ M, q_1 \models i \langle \langle 1 \rangle \rangle \circ ((\langle \langle 2 \rangle \rangle \circ \langle \langle \emptyset \rangle \rangle \circ \neg p) \land \langle \langle 2 \rangle \rangle \circ \langle \langle \emptyset \rangle \rangle \circ p) \]

(strategies: \( \{ q_3 \mapsto \alpha_1, q_5 \mapsto \alpha_2 \} ; \{ q_2 \mapsto \beta_1 \} ; \{ q_2 \mapsto \beta_2 \} \) )

\[ M', q_1 \not\models i \langle \langle 1 \rangle \rangle \circ ((\langle \langle 2 \rangle \rangle \circ \langle \langle \emptyset \rangle \rangle \circ \neg p) \land \langle \langle 2 \rangle \rangle \circ \langle \langle \emptyset \rangle \rangle \circ p) \]
On Valid Reasoning about Irrevocable Strategies

- Formulae valid under the standard definition is not necessarily valid under irrevocable strategies. For example, the principle of uniform substitution does not hold. The ATL axiom

\[ \neg \langle \langle \emptyset \rangle \rangle \bigcirc \neg p \rightarrow \langle \langle \Sigma \rangle \rangle \bigcirc p \]

is still valid with irrevocable strategies, but the result of substituting

\[ \langle \langle \Sigma \rangle \rangle \bigcirc p \land \langle \langle \Sigma \rangle \rangle \bigcirc \neg p \]

for \( p \) in it is not valid.
On Valid Reasoning about Irrevocable Strategies

- Formulae valid under the standard definition is not necessarily valid under irrevocable strategies. For example, the principle of uniform substitution does not hold. The ATL axiom

\[ \neg \langle \langle \emptyset \rangle \rangle \bigcirc \neg p \rightarrow \langle \langle \Sigma \rangle \rangle \bigcirc p \]

is still valid with irrevocable strategies, but the result of substituting

\[ \langle \langle \Sigma \rangle \rangle \bigcirc p \land \langle \langle \Sigma \rangle \rangle \bigcirc \neg p \]

for \( p \) in it is not valid.

- Formulae valid under irrevocable strategies are not necessarily valid under the standard definition. Example:

\[ \langle \langle C \rangle \rangle \bigcirc \langle \langle C' \rangle \rangle \bigcirc \phi \leftrightarrow \langle \langle C' \rangle \rangle \bigcirc \langle \langle \emptyset \rangle \rangle \bigcirc \phi \]
Perfect Recall

With perfect recall strategies, we cannot update the model directly. Instead, unwind it first, and recall that a perfect recall strategy in $\mathcal{M}$ is equivalent to a memoryless strategy in $T(\mathcal{M}, q)$:

$$\mathcal{M}, q \models_{mi} \varphi \iff \text{def} \ T(\mathcal{M}, q), q \models_{i} \varphi$$
Perfect Recall

\[ \mathcal{M}, q \models_{mi} \varphi \iff \text{def} T(M, q), q \models_i \varphi \]
Perfect Recall

\[ \mathcal{M}, q \models_{mi} \varphi \iff \overset{\text{def}}{\Rightarrow} T(M, q), q \models_{i} \varphi \]

We get that:

- Still non-invariant under bisimulation
Perfect Recall

\[ \mathcal{M}, q \models_{mi} \varphi \iff \text{def } T(M, q), q \models_i \varphi \]

We get that:
- Still non-invariant under bisimulation
- With irrevocable strategies (unlike under the standard definition), memory matters:

\[ \mathcal{M}, q_1 \models_{mi} \langle \langle a \rangle \rangle \circ \langle \langle a \rangle \rangle \circ p \]
\[ \mathcal{M}, q_1 \not\models_i \langle \langle a \rangle \rangle \circ \langle \langle a \rangle \rangle \circ p \]
4. More about ATL

4.4 References
4. More about ATL


How can we reason about extensive games with **imperfect information**?
5. Imperfect Information

How can we reason about extensive games with imperfect information?

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

- We extend CGS with indistinguishability relations $\sim_a$, one per agent
- We add epistemic operators to ATL
5. Imperfect Information

How can we reason about extensive games with imperfect information?

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

- We extend CGS with indistinguishability relations $\sim_a$, one per agent
- We add epistemic operators to ATL

$\leadsto$ Problems!
5.1 Combining Dimensions
5. Imperfect Information

1. Combining Dimensions

---

Does it make sense?
5. Imperfect Information

1. Combining Dimensions

\[
\begin{align*}
q_0 &\quad \text{start} \\
(q_1, q_2, q_3, q_4, q_5) &\quad \text{keep, trade, keep, keep, keep} \\
(q_6) &\quad \text{trade, trade, trade, trade, trade} \\
\end{align*}
\]
Does it make sense?

\( \text{start} \rightarrow \langle \langle a \rangle \rangle \diamond \text{win} \)
Does it make sense?

\[\text{start} \rightarrow \langle \langle a \rangle \rangle \Diamond \text{win} \]

\[\text{start} \rightarrow K_a \langle \langle a \rangle \rangle \Diamond \text{win} \]
Does it make sense?
Problem:
Strategic and epistemic abilities are not independent!
Problem:
Strategic and epistemic abilities are not independent!

$$\langle A \rangle \Phi = A \text{ can enforce } \Phi$$
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!
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 5.1 (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 $i$. 
Definition 5.1 (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!
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

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

1. There is $\sigma$ (not necessarily executable!) such that, for every execution of $\sigma$, $\Phi$ holds
Levels of Strategic Ability

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

1. There is $\sigma$ (not necessarily executable!) such that, for every execution of $\sigma$, $\Phi$ holds
2. There is a uniform $\sigma$ such that, for every execution of $\sigma$, $\Phi$ holds
Levels of Strategic Ability

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

1. There is $\sigma$ (not necessarily executable!) such that, for every execution of $\sigma$, $\Phi$ holds
2. There is a uniform $\sigma$ such that, for every execution of $\sigma$, $\Phi$ holds
3. $A$ know that there is a uniform $\sigma$ such that, for every execution of $\sigma$, $\Phi$ holds
Levels of Strategic Ability

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

1. There is $\sigma$ (not necessarily executable!) such that, for every execution of $\sigma$, $\Phi$ holds
2. There is a uniform $\sigma$ such that, for every execution of $\sigma$, $\Phi$ holds
3. $A$ know that there is a uniform $\sigma$ such that, for every execution of $\sigma$, $\Phi$ holds
4. There is a uniform $\sigma$ such that $A$ know that, for every execution of $\sigma$, $\Phi$ holds
Levels of Strategic Ability

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

1. There is $\sigma$ (not necessarily executable!) such that, for every execution of $\sigma$, $\Phi$ holds
2. There is a uniform $\sigma$ such that, for every execution of $\sigma$, $\Phi$ holds
3. $A$ know that there is a uniform $\sigma$ such that, for every execution of $\sigma$, $\Phi$ holds
4. There is a uniform $\sigma$ such that $A$ know that, for every execution of $\sigma$, $\Phi$ holds

From now on, we restrict our discussion to uniform memoryless strategies (unless explicitly stated otherwise).
Levels of Strategic Ability

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

1. There is $\sigma$ (not necessarily executable!) such that, for every execution of $\sigma$, $\Phi$ holds
2. There is a uniform $\sigma$ such that, for every execution of $\sigma$, $\Phi$ holds
3. $A$ know that there is a uniform $\sigma$ such that, for every execution of $\sigma$, $\Phi$ holds
4. There is a uniform $\sigma$ such that $A$ know that, for every execution of $\sigma$, $\Phi$ holds

From now on, we restrict our discussion to uniform memoryless strategies (unless explicitly stated otherwise).
Case [4]: knowing how to play
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), mutual knowledge (E), distributed knowledge (D)?
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 (\( E_A \)), distributed knowledge (\( D_A \))?
5.2 Economic Solution: \(\text{ATL}_{ir}\)
Schobbens’ ATL$_{ir}$

$\langle \mathcal{A} \rangle_{ir}^\gamma$: agents $\mathcal{A}$ know how to play in the sense of mutual knowledge ($E_{\mathcal{A}}$)
Schobbens’ ATL$_{ir}$

$\langle A \rangle_{ir} \gamma$: agents $A$ know how to play in the sense of mutual knowledge ($E_A$)

$M, q \models \langle A \rangle_{ir} \gamma$ iff there is a collective uniform strategy $s_A$ such that, for every path $\lambda \in \bigcup_{q' \sim A q} \text{out}(q', s_A)$, we have $M, \Lambda \models \gamma$. 
Example: Robots and Carriage
Example: Robots and Carriage

The diagram illustrates the states and transitions between them, where:
- States: $q_0$, $q_1$, $q_2$
- Positions: $pos_0$, $pos_1$, $pos_2$
- Transitions:
  - From $q_0$ to $q_1$: $wait,wait$, $push,push$
  - From $q_0$ to $q_2$: $push,wait$, $wait,push$
  - From $q_1$ to $q_0$: $wait,push$
  - From $q_1$ to $q_2$: $push,wait$
  - From $q_2$ to $q_0$: $wait,wait$, $push,push$
  - From $q_2$ to $q_1$: $push,wait$

The numbers 1 and 2 indicate different scenarios or paths through the system.
Example: Robots and Carriage

\[
\neg (\text{pos}_0 \rightarrow \langle \langle s \rangle \rangle_{ir} \square \neg \text{pos}_0)
\]

\[
\text{pos}_0 \rightarrow \langle \langle s \rangle \rangle_{ir} \square \neg \text{pos}_1
\]
5. Imperfect Information

Schobbens’ ATL\(_{ir}\)

Interesting: \(\langle A \rangle_{ir}\) are not fixpoint operators any more!

**Theorem 5.2**

The following formulae are not valid for ATL\(_{ir}\):

- \(\langle A \rangle_{ir} \Box \varphi \leftrightarrow \varphi \land \langle A \rangle_{ir} \bigcirc \langle A \rangle_{ir} \Box \varphi\)
- \(\langle A \rangle_{ir} \varphi_1 U \varphi_2 \leftrightarrow \varphi_2 \lor \varphi_1 \land \langle A \rangle_{ir} \bigcirc \langle A \rangle_{ir} \varphi_1 U \varphi_2\).
Schobbens’ ATL$_{ir}$

Interesting: $\langle \langle A \rangle \rangle_{ir}$ are not fixpoint operators any more!

**Theorem 5.2**

The following formulae are not valid for ATL$_{ir}$:

- $\langle \langle A \rangle \rangle_{ir} \Box \varphi \iff \varphi \land \langle \langle A \rangle \rangle_{ir} \bigcirc \langle \langle A \rangle \rangle_{ir} \Box \varphi$

- $\langle \langle A \rangle \rangle_{ir} \varphi_1 U \varphi_2 \iff \varphi_2 \lor \varphi_1 \land \langle \langle A \rangle \rangle_{ir} \bigcirc \langle \langle A \rangle \rangle_{ir} \varphi_1 U \varphi_2$.  

What is it about?
Schobbens’ ATL$_{ir}$

Interesting: $\langle A \rangle_{ir}$ are not fixpoint operators any more!

**Theorem 5.2**

The following formulae are **not** valid for ATL$_{ir}$:

- $\langle A \rangle_{ir} \square \varphi \iff \varphi \land \langle A \rangle_{ir} \bigcirc \langle A \rangle_{ir} \square \varphi$
- $\langle A \rangle_{ir} \varphi_1 U \varphi_2 \iff \varphi_2 \lor \varphi_1 \land \langle A \rangle_{ir} \bigcirc \langle A \rangle_{ir} \varphi_1 U \varphi_2$

What is it about? **Forgetting!**
Agents Can Forget...
Agents Can Forget... And Still Enforce Things
Schobbens’ $\text{ATL}_{ir}$

**Conjecture**

Strategy for $A$ cannot be synthesized incrementally.
Schobbens’ ATL$_{ir}$

**Conjecture**

Strategy for $A$ cannot be synthesized incrementally.

Indeed...
Schobbens’ ATL$_{ir}$

**Conjecture**
Strategy for $A$ cannot be synthesized incrementally.

Indeed...

**Theorem (Schobbens 2004; Jamroga & Dix 2006)**
Model checking ATL$_{ir}$ is $\Delta_2$-complete in the number of transitions in the model and the length of the formula.
5.3 Constructive Strategic Logic
Knowing how to Play

- Single agent case: we take into account the paths starting from indistinguishable states \( \leadsto ATL_{ir} \)

- What about coalitions? **In what sense** should they know the strategy? Common knowledge \( (C_A) \), mutual knowledge \( (E_A) \), distributed knowledge \( (D_A) \)...?

- ATL\(_{ir}\): mutual knowledge

- But: other cases also make sense!
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).
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
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.
- 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 $a \in A$. 

- 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 $A' \subseteq A$. 
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...
Many subtle cases...

~~ Solution: constructive knowledge operators
Constructive Strategic Logic (CSL)

- $\langle A \rangle \Phi$: A have a uniform memoryless strategy to enforce $\Phi$
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: $C_A, E_A, D_A, ...$
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: $C_A, E_A, D_A, ...$
- $K_a \langle a \rangle \Phi$: $a$ 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
- $M, Q \models \langle A \rangle \gamma$: $A$ have a **single** strategy to enforce $\gamma$ from all states in $Q$
Non-standard semantics:

- Formulae are evaluated in sets of states
- $M, Q \models \ll A \gg \gamma$: A have a single strategy to enforce $\gamma$ from all states in $Q$

Additionally:

- $\text{out}(Q, s_A) = \bigcup_{q \in Q} \text{out}(q, s_A)$
- $\text{img}(Q, \mathcal{R}) = \bigcup_{q \in Q} \text{img}(q, \mathcal{R})$
Non-standard semantics:

- Formulae are evaluated in sets of states
- $M, Q \models \langle A \rangle \gamma$: $A$ have a single strategy to enforce $\gamma$ from all states in $Q$

Additionally:

- $\text{out}(Q, s_A) = \bigcup_{q \in Q} \text{out}(q, s_A)$
- $\text{img}(Q, \mathcal{R}) = \bigcup_{q \in Q} \text{img}(q, \mathcal{R})$

- $M, q \models \varphi$ iff $M, \{q\} \models \varphi$
Definition 5.3 (Semantics of CSL)

\[ M, Q \models p \quad \text{iff} \quad p \in \pi(q) \text{ for every } q \in Q; \]
5. Imperfect Information

3. Constructive Strategic Logic

Definition 5.3 (Semantics of CSL)

\[ M, Q \models p \quad \text{iff} \quad p \in \pi(q) \text{ for every } q \in Q; \]
\[ M, Q \models \neg \varphi \quad \text{iff} \quad \text{not } M, Q \models \varphi; \]
Definition 5.3 (Semantics of CSL)

\[ M, Q \models p \iff p \in \pi(q) \text{ for every } q \in Q; \]

\[ M, Q \models \neg \varphi \iff \text{not } M, Q \models \varphi; \]

\[ M, Q \models \varphi \land \psi \iff M, Q \models \varphi \text{ and } M, Q \models \psi; \]
5. Imperfect Information

Definition 5.3 (Semantics of CSL)

\[ M, Q \models p \quad \text{iff} \quad p \in \pi(q) \text{ for every } q \in Q; \]
\[ M, Q \models \neg \varphi \quad \text{iff} \quad \text{not } M, Q \models \varphi; \]
\[ M, Q \models \varphi \land \psi \quad \text{iff} \quad M, Q \models \varphi \text{ and } M, Q \models \psi; \]
\[ M, Q \models \langle A \rangle \gamma \quad \text{iff} \quad \text{there exists } s_A \text{ such that, for every } \lambda \in \text{out}(Q, s_A), \text{ we have that } M, \lambda \models \gamma; \]
**Definition 5.3 (Semantics of CSL)**

\[ M, Q \models p \quad \text{iff} \quad p \in \pi(q) \text{ for every } q \in Q; \]

\[ M, Q \models \neg \varphi \quad \text{iff not } \quad M, Q \models \varphi; \]

\[ M, Q \models \varphi \land \psi \quad \text{iff } \quad M, Q \models \varphi \text{ and } M, Q \models \psi; \]

\[ M, Q \models \langle A \rangle \gamma \quad \text{iff there exists } s_A \text{ such that, for every } \lambda \in \text{out}(Q, s_A), \text{ we have that } M, \lambda \models \gamma; \]

\[ M, Q \models \mathcal{K}_A \varphi \quad \text{iff } M, q \models \varphi \text{ for every } q \in \text{img}(Q, \sim^\mathcal{K}_A) \text{ (where } \mathcal{K} = C, E, D); \]
Definition 5.3 (Semantics of CSL)

\[ M, Q \models p \iff p \in \pi(q) \text{ for every } q \in Q; \]
\[ M, Q \models \neg \varphi \iff \text{not } M, Q \models \varphi; \]
\[ M, Q \models \varphi \land \psi \iff M, Q \models \varphi \text{ and } M, Q \models \psi; \]
\[ M, Q \models \langle A \rangle \gamma \iff \text{there exists } s_A \text{ such that, for every } \lambda \in \text{out}(Q, s_A), \text{ we have that } M, \lambda \models \gamma; \]
\[ M, Q \models K_A \varphi \iff M, q \models \varphi \text{ for every } q \in \text{img}(Q, \sim^K_A) \text{ (where } K = C, E, D); \]
\[ M, Q \models \hat{K}_A \varphi \iff M, \text{img}(Q, \sim^K_A) \models \varphi \text{ (where } \hat{K} = C, E, D \text{, respectively).} \]
Validity in CSL

- Formula $\phi$ is **valid** iff $M, q \models \phi$ for all models $M$ and states $q$.
- Formula $\phi$ is **strongly valid** iff for each $M$ and every non-empty set of states $Q$ it is the case that $M, Q \models \phi$.
Validity in CSL

- Formula $\varphi$ is **valid** iff $M, q \models \varphi$ for all models $M$ and states $q$
- Formula $\varphi$ is **strongly valid** iff for each $M$ and every non-empty set of states $Q$ it is the case that $M, Q \models \varphi$

**Theorem 5.4**

1. *Strong validity implies validity.*
2. *Validity does not imply strong validity.*
Validity in CSL

- We are ultimately interested in simple validity
Validity in CSL

- We are ultimately interested in simple validity.
- The importance of strong validity, on the other hand, lies in the fact that strong validity of $\varphi \leftrightarrow \psi$ makes $\varphi$ and $\psi$ completely interchangeable.

\[ \text{Theorem 5.5} \]

If $\varphi_1 \leftrightarrow \varphi_2$ is strongly valid, and $\psi'$ is obtained from $\psi$ through replacing an occurrence of $\varphi_1$ by $\varphi_2$, then $M,Q \models \psi$ iff $M,Q \models \psi'$. 
Validity in CSL

- We are ultimately interested in simple validity
- The importance of strong validity, on the other hand, lies in the fact that strong validity of $\varphi \leftrightarrow \psi$ makes $\varphi$ and $\psi$ completely interchangeable

**Theorem 5.5**

If $\varphi_1 \leftrightarrow \varphi_2$ is strongly valid, and $\psi'$ is obtained from $\psi$ through replacing an occurrence of $\varphi_1$ by $\varphi_2$, then $M, Q \models \psi$ iff $M, Q \models \psi'$. 
5. Imperfect Information

3. Constructive Strategic Logic

Example: Simple Market

@ $q_1$: $\neg \mathcal{K}_c(c) \diamond \text{success}$
Example: Simple Market

@ $q_1$:

\[\neg K_c \langle c \rangle \Diamond \text{success} \]

\[\neg E_{\{1,2\}} \langle c \rangle \Diamond \text{success} \]

\[\neg K_1 \langle c \rangle \Diamond \text{success} \]

\[\neg K_2 \langle c \rangle \Diamond \text{success} \]
Example: Simple Market

@ \( q_1 \):

- \( \neg K_c \langle \langle c \rangle \rangle \diamond \text{success} \)
- \( \neg E \{1,2\} \langle \langle c \rangle \rangle \diamond \text{success} \)
- \( \neg K_1 \langle \langle c \rangle \rangle \diamond \text{success} \)
- \( \neg K_2 \langle \langle c \rangle \rangle \diamond \text{success} \)
- \( \mathcal{D} \{1,2\} \langle \langle c \rangle \rangle \diamond \text{success} \)
Onion Soup Robbery

A virtual safe contains the recipe for the best onion soup in the world. The safe can only be opened by a $k$-digit binary code, where each digit $c_i$ is sent from a prescribed location $i$ ($1 \leq i \leq k$). To open the safe and download the recipe it is enough that at least $n \leq k$ correct digits are sent at the same moment. However, if a wrong value is sent from one of the locations, or if an insufficient number (i.e., between 1 and $n-1$) of digits is submitted, then the safe locks up and activates an alarm.

$k$ agents are connected at the right locations; each of them can send 0, send 1, or do nothing ($nop$). Moreover, individual agents have only partial information about the code: agent $i$ (connected to location $i$) knows the values of $c_{i-1} \oplus c_i$ and $c_i \oplus c_{i+1}$ (we take $c_0 = c_{k+1} = 0$). This implies that only agents 1 and $k$ know the values of “their” digits. Still, every agent knows whether his neighbors’ digits are the same as his.
Onion Soup Robbery: Some Properties

For $OSR^n_k$ and the initial state, we have:

- $\neg E_{\text{Agt}} \langle \langle \text{Agt} \rangle \rangle \diamond \text{open}$: the team cannot identify a winning strategy;
Onion Soup Robbery: Some Properties

For $OSR^n_k$ and the initial state, we have:

- $\neg E_{\text{Agt}} \langle \langle \text{Agt} \rangle \rangle \diamond \text{open}$: the team cannot identify a winning strategy;

- $D_{\text{Agt}} \langle \langle \text{Agt} \rangle \rangle \diamond \text{open}$: if the agents share information they can recognize who should send what;
Onion Soup Robbery: Some Properties

For $OSR^n_k$ and the initial state, we have:

- $\neg \exists_{Agt} \langle Agt \rangle \Diamond \text{open}$: the team cannot identify a winning strategy;

- $D_{Agt} \langle Agt \rangle \Diamond \text{open}$: if the agents share information they can recognize who should send what;

- $D_{\{1,...,n-1\}} \langle Agt \rangle \Diamond \text{open}$: it is enough that the first $n - 1$ agents devise the strategy. Note that the same holds for the last $n - 1$ agents, i.e., the subteam $\{k - n + 2, \ldots, k\}$.
Theorem 5.6 (Expressivity)

CSL is strictly more expressive than ATL_{ir}.
Theorem 5.6 (Expressivity)

{
CSL is strictly more expressive than ATL$_{ir}$.

Theorem 5.7 (Verification complexity)

The complexity of model checking CSL is the same as for ATL$_{ir}$.
5.4 Constructive Knowledge
Properties of Constructive Knowledge

Non-standard semantics raises some natural questions:

- Is constructive knowledge... em, well, knowledge?
5. Imperfect Information

4. Constructive Knowledge

Properties of Constructive Knowledge

Non-standard semantics raises some natural questions:

- Is constructive knowledge... em, well, knowledge?
- ~ semantic vs. syntactic analysis
Properties of Constructive Knowledge

Non-standard semantics raises some natural questions:

- Is constructive knowledge... em, well, knowledge?
  \[\sim\] semantic vs. syntactic analysis

- Is constructive knowledge a special kind of standard knowledge? Or the other way around?
Properties of Constructive Knowledge

Non-standard semantics raises some natural questions:

- Is constructive knowledge... em, well, knowledge?
  $\leadsto$ semantic vs. syntactic analysis

- Is constructive knowledge a special kind of standard knowledge? Or the other way around?

- Is there a relevant subset of the language for whom a more standard semantics can be given?
Is $K_a$ an Epistemic Operator?

**Theorem 5.8**

Below, we list the constructive knowledge versions of some of the S5 properties for individual agents. “Yes” means that the schema is strongly valid; “No” means that it is not even weakly valid (incidentally, none of the properties turns out to be weakly but not strongly valid).

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Validity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_a$</td>
<td>$K_a (\varphi \rightarrow \psi) \rightarrow (K_a \varphi \rightarrow K_a \psi)$</td>
<td>Yes</td>
</tr>
<tr>
<td>$D$</td>
<td>$\neg K_a \bot$</td>
<td>Yes</td>
</tr>
<tr>
<td>$T$</td>
<td>$K_a \varphi \rightarrow \varphi$</td>
<td>No</td>
</tr>
<tr>
<td>$4$</td>
<td>$K_a \varphi \rightarrow K_a K_a \varphi$</td>
<td>Yes</td>
</tr>
<tr>
<td>$4^+$</td>
<td>$K_a \varphi \leftrightarrow K_a K_a \varphi$</td>
<td>Yes</td>
</tr>
<tr>
<td>$5$</td>
<td>$\neg K_a \varphi \rightarrow K_a \neg K_a \varphi$</td>
<td>Yes</td>
</tr>
<tr>
<td>$5^+$</td>
<td>$\neg K_a \varphi \leftrightarrow K_a \neg K_a \varphi$</td>
<td>Yes</td>
</tr>
<tr>
<td>$B$</td>
<td>$\varphi \rightarrow K_a \neg K_a \neg \varphi$</td>
<td>No</td>
</tr>
</tbody>
</table>
Is $K_a$ an Epistemic Operator?

**Theorem 5.8**

Below, we list the constructive knowledge versions of some of the S5 properties for individual agents. “Yes” means that the schema is strongly valid; “No” means that it is not even weakly valid (incidentally, none of the properties turns out to be weakly but not strongly valid).

<table>
<thead>
<tr>
<th>Property</th>
<th>Schema</th>
<th>Validity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K$</td>
<td>$K_a(\varphi \rightarrow \psi) \rightarrow (K_a\varphi \rightarrow K_a\psi)$</td>
<td>Yes</td>
</tr>
<tr>
<td>$D$</td>
<td>$\neg K_a \bot$</td>
<td>Yes</td>
</tr>
<tr>
<td>$T$</td>
<td>$K_a\varphi \rightarrow \varphi$</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>$K_a\varphi \rightarrow K_aK_a\varphi$</td>
<td>Yes</td>
</tr>
<tr>
<td>4$^+$</td>
<td>$K_a\varphi \leftrightarrow K_aK_a\varphi$</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>$\neg K_a\varphi \rightarrow K_a\neg K_a\varphi$</td>
<td>Yes</td>
</tr>
<tr>
<td>5$^+$</td>
<td>$\neg K_a\varphi \leftrightarrow K_a\neg K_a\varphi$</td>
<td>Yes</td>
</tr>
<tr>
<td>$B$</td>
<td>$\varphi \rightarrow K_a\neg K_a\neg \varphi$</td>
<td>No</td>
</tr>
</tbody>
</table>
Invalidity of Axiom T

Let $M$ be as above
Now, $M, q \models \Box a \neg p$, but $M, q \not\models \neg p$
In Quest for the Truth Axiom

- $K_a$ is not S5: axioms $K, D, 4, 5$ hold, but $T$ does not
- However, if we slightly restrict the language, then the corresponding $T$ axiom becomes strongly valid
In Quest for the Truth Axiom

- $K_a$ is not S5: axioms $K, D, 4, 5$ hold, but $T$ does not
- However, if we slightly restrict the language, then the corresponding $T$ axiom becomes strongly valid

- Let $\text{CSL}^-$ be the subset of CSL in which, between every occurrence of constructive knowledge ($C_A, E_A, D_A$) and negation, there is always at least one operator other than conjunction
- In particular, the requirement is met when $C_A, E_A, D_A$ are never immediately followed by $\neg$ or $\land$
In Quest for the Truth Axiom

- $K_a$ is not S5: axioms $K$, $D$, 4, 5 hold, but $T$ does not
- However, if we slightly restrict the language, then the corresponding $T$ axiom becomes strongly valid

- Let CSL$^-$ be the subset of CSL in which, between every occurrence of constructive knowledge ($C_A$, $E_A$, $D_A$) and negation, there is always at least one operator other than conjunction
- In particular, the requirement is met when $C_A$, $E_A$, $D_A$ are never immediately followed by $\neg$ or $\land$

**Theorem 5.9**

Every CSL$^-$ instance of $T$ (i.e., $K_a \psi \rightarrow \psi$) is strongly valid.
In Quest for the Truth Axiom

Is then the constructive knowledge in CSL\(^{-}\) S5?
In Quest for the Truth Axiom

Is then the constructive knowledge in CSL⁻ S5?
Not really
In Quest for the Truth Axiom

Is then the constructive knowledge in CSL$^-$ S5?

Not really

- The extension of schema $T$ is **different** in CSL and CSL$^-$
- More importantly, in CSL$^-$ schemata $K$ and 5 are not valid, but they are not invalid either – they are simply not formulae at all
- Finally, CSL$^-$ lacks the S5 principle of **uniform substitution**
Properties of Collective Constructive Knowledge

**Theorem 5.10**

Below, we list some of the S5 properties for collective constructive knowledge operators. “Yes” means that the schema is strongly valid; “No” means that it is not even weakly valid.

<table>
<thead>
<tr>
<th></th>
<th>$C_A$</th>
<th>$E_A$</th>
<th>$D_A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>D</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>T</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>$4^+$</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>$5^+$</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>B</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Properties of Collective Constructive Knowledge

**Theorem 5.11**

*Every CSL⁻ instance of schema $T$ for collective constructive knowledge operators $C_A, E_A, D_A$ is strongly valid.*
Normal Form and State-Based Semantics

Constructive Normal Form

A CSL formula is in constructive normal form (CSNF) if every subformula starting with a $\hat{\mathcal{K}}_A$ operator is of the form $\hat{\mathcal{K}}_{A_1} \ldots \hat{\mathcal{K}}_{A_n} \psi$ where $\psi$ starts with a cooperation modality.
Normal Form and State-Based Semantics

Constructive Normal Form

A CSL formula is in constructive normal form (CSNF) if every subformula starting with a \( \hat{\mathcal{K}}_A \) operator is of the form \( \hat{\mathcal{K}}_{A_1} \ldots \hat{\mathcal{K}}_{A_n} \psi \) where \( \psi \) starts with a cooperation modality.

Proposition

Every CSL formula is strongly equivalent to a formula in constructive normal form.

Note: equivalent does not mean the same!
Normal Form CSL

Observation

The “normal form CSL” can be given semantics entirely in terms of models and states.
Normal Form CSL

**Observation**

The “normal form CSL” can be given semantics entirely in terms of models and states.

\[ M, q \models \hat{K}^1_{A_1} \ldots \hat{K}^n_{A_n} \langle A \rangle \gamma \iff \text{there exists } S_A \text{ such that, for every } \lambda \in \text{out}(\text{img}(q, rel(\hat{K}^1_{A_1} \ldots \hat{K}^n_{A_n}), S_A)), \text{ we have that } M, \lambda \models \gamma, \]

where \( rel(\hat{K}^1_{A_1} \ldots \hat{K}^n_{A_n}) = \sim_{A_1}^1 \circ \ldots \circ \sim_{A_n}^n. \)
Normal Form CSL vs. Onion Soup

- $\neg E_{\text{Agt}}\langle \langle \text{Agt} \rangle \rangle \diamond \text{open}$
- $D_{\text{Agt}}\langle \langle \text{Agt} \rangle \rangle \diamond \text{open}$
- $D\{1,...,n-1\}\langle \langle \text{Agt} \rangle \rangle \diamond \text{open}$

These are normal form formulae!
5.5 Between Perception and Recall
Strategies for Different Settings

Four variants of ability: IR, Ir, iR, ir (Schobbens 2004)

- I/i: perfect/imperfect information
- R/r: perfect/imperfect recall
Strategies for Different Settings

Four variants of ability: IR, Ir, iR, ir (Schobbens 2004)
- I/i: perfect/imperfect information
- R/r: perfect/imperfect recall
- r: \( s_a : S^t \rightarrow Act \) (memoryless strategies)
- R: \( s_a : S^t+ \rightarrow Act \) (perfect recall strategies)
Strategies for Different Settings

Four variants of ability: IR, Ir, iR, ir (Schobbens 2004)

- **I/i**: perfect/imperfect **information**
- **R/r**: perfect/imperfect **recall**

- **r**: \( s_a : St \rightarrow Act \) (memoryless strategies)
- **R**: \( s_a : St^+ \rightarrow Act \) (perfect recall strategies)

- **i**: only uniform strategies,
- **l**: no restrictions
5. Imperfect Information

5. Between Perception and Recall

Strategies for Different Settings

Four variants of ability: IR, Ir, iR, ir (Schobbens 2004)

- I/i: perfect/imperfect information
- R/r: perfect/imperfect recall
- r: $s_a : St \rightarrow \text{Act}$ (memoryless strategies)
- R: $s_a : St^+ \rightarrow \text{Act}$ (perfect recall strategies)
- i: only uniform strategies,
- I: no restrictions

- r: $s_a$ is uniform iff $q \sim a q' \Rightarrow s_a(q) = s_a(q')$
- R: $s_a$ is uniform iff $\lambda \approx a \lambda' \Rightarrow s_a(\lambda) = s_a(\lambda')$
- $\lambda \approx a \lambda'$ iff $\forall i \lambda[i] \sim a \lambda'[i]$
Model Checking Complexity

<table>
<thead>
<tr>
<th>Logic</th>
<th>$ir$</th>
<th>$iR$</th>
<th>$Ir$</th>
<th>$IR$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{ATL}^+$</td>
<td>$\Delta_3P$</td>
<td>$U[11]$</td>
<td>$\Delta_3P$</td>
<td>$\Delta_3P$</td>
</tr>
<tr>
<td>$\text{ATL}^*$</td>
<td>$\text{PSPACE}$</td>
<td>$U[11]$</td>
<td>$\text{PSPACE}$</td>
<td>$\text{DEXP}[9]$</td>
</tr>
</tbody>
</table>

$NP$ complete for nondeterministic polynomial time

$\Delta_2P = P^{NP}$ complete for polynomial calls to an $NP$ oracle

$\Delta_3P = P^{NP^{NP}}$ complete for polynomial calls to a $\Sigma_2P$ oracle

$\text{EXP}$ complete for deterministic exponential time

$\text{DEXP}$ complete for deterministic doubly exponential time

$U$ undecidable

$l$ size of the formula

$n$ size of the model
Perfect vs. Imperfect Recall

- Diagram illustrating the transitions between states with actions like "wait, wait" and "push, push".
- States labeled q₀, q₁, and q₂ with positions pos₀, pos₁, and pos₂.

Advice: The restrictions on strategies and the semantics of epistemic operators should match!
Perfect vs. Imperfect Recall

(A)

Thomas Ågotnes and Wojtek Jamroga · Modal Logics for Games and MAS

ESSLLI 2008 @ Hamburg

181/355
Perfect vs. Imperfect Recall

(A)

(B)
Perfect vs. Imperfect Recall

Advice: the restrictions on strategies and the semantics of epistemic operators should match!
5.6 References
5. Imperfect Information

Alternating-time logic with imperfect recall.  

Constructive knowledge: What agents can achieve under incomplete information.  

Cooperation, knowledge and time: Alternating-time Temporal Epistemic Logic and its applications.  

Introduction

- Let us look at how we can logically characterise solution concepts for strategic games.
Introduction

- Let us look at how we can logically characterise solution concepts for strategic games
- Modal logic characterisations of solution concepts have been studied by many authors, e.g.
  - Bonanno: both strategic and extensive games
  - Harrenstein et al.: extensive form games; modalities for preferences (see our Friday lectures)
- Here: we will take Coalition Logic/ATL as a starting point

- **Strategic game:** $G = (N, \{\Sigma_i : i \in N\}, o, S, \{\succeq_i : i \in N\})$
- **Solution concepts** describe outcomes of games
Nash Equilibrium

Informally: given a game, a strategy profile is a Nash equilibrium iff every strategy is a best response (for that agent) to the other strategies.

Nash Equilibrium

Informally: given a game, a strategy profile is a Nash equilibrium iff every strategy is a best response (for that agent) to the other strategies.

Formally:

\[ G = (N, \{\Sigma_i : i \in N\}, o, S, \{\succeq_i : i \in N\}) \]

**Definition 6.1 (Nash Equilibrium)**

A strategy profile \( \sigma_N \) is a (pure strategy) Nash equilibrium of \( G \) iff for every \( i \in N \) and \( \sigma'_i \)

\[ o(\sigma_i, \sigma_{-i}) \succeq_i o(\sigma'_i, \sigma_{-i}) \]
### Nash Equilibrium: example: Prisoner’s dilemma

<table>
<thead>
<tr>
<th>Ann</th>
<th>Bill</th>
<th>Cooperate</th>
<th>Defect</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cooperate</td>
<td>Ann:-1, Bill:-1</td>
<td>Ann:-4, Bill: 0</td>
</tr>
<tr>
<td></td>
<td>Defect</td>
<td>Ann:0, Bill:-4</td>
<td>Ann:-3, Bill: -3</td>
</tr>
</tbody>
</table>
### Nash Equilibrium: example: Prisoner’s dilemma

<table>
<thead>
<tr>
<th></th>
<th>Cooperate</th>
<th>Defect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cooperate</td>
<td>Ann:-1, Bill:-1</td>
<td>Ann:-4, Bill: 0</td>
</tr>
<tr>
<td>Defect</td>
<td>Ann:0, Bill:-4</td>
<td>Ann:-3, Bill: -3</td>
</tr>
</tbody>
</table>

One Nash equilibrium: (Defect, Defect)
### Nash Equilibrium: example: Bach or Stravinsky

<table>
<thead>
<tr>
<th></th>
<th>Bill B</th>
<th>Bill S</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ann B</td>
<td>Ann:2, Bill:1</td>
<td>Ann:0, Bill:0</td>
</tr>
<tr>
<td>Ann S</td>
<td>Ann:0, Bill:0</td>
<td>Ann:1, Bill:2</td>
</tr>
</tbody>
</table>
Nash Equilibrium: example: Bach or Stravinsky

<table>
<thead>
<tr>
<th>Ann</th>
<th>B</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Ann:2, Bill:1</td>
<td>Ann:0, Bill: 0</td>
</tr>
<tr>
<td>S</td>
<td>Ann:0, Bill:0</td>
<td>Ann:1, Bill: 2</td>
</tr>
</tbody>
</table>

Two Nash equilibria: (B,B) and (S,S)
Weakly Dominant Strategies

Informally: a strategy is weakly dominant if it is as least as good as any other strategy no matters what the other agents do.

Weakly Dominant Strategies

Informally: a strategy is weakly dominant if it is as least as good as any other strategy no matters what the other agents do.

Formally:

Definition 6.2 (Weak Dominance)

A strategy $\sigma_i$ weakly dominates strategy $\sigma'_i$ iff for all $\sigma_{-i}$

$$o(\sigma_i, \sigma_{-i}) \geq_i o(\sigma'_i, \sigma_{-i})$$

A strategy is weakly dominant for $i$ iff it weakly dominates all other strategies for $i$. 
Dominance: example: Prisoner’s dilemma

<table>
<thead>
<tr>
<th></th>
<th>Ann: 0, Bill: -4</th>
<th>Ann: -3, Bill: -3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Defect</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Cooperate</strong></td>
<td>Ann: -1, Bill: -1</td>
<td>Ann: -4, Bill: 0</td>
</tr>
</tbody>
</table>

Bill

Dominance: example: Prisoner’s dilemma
Dominance: example: Prisoner’s dilemma

<table>
<thead>
<tr>
<th></th>
<th>Bill</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cooperate</td>
</tr>
<tr>
<td>Ann</td>
<td>Cooperate</td>
</tr>
<tr>
<td></td>
<td>Defect</td>
</tr>
</tbody>
</table>

**Defect** is dominant for Ann
### Dominance: example: Prisoner’s dilemma

<table>
<thead>
<tr>
<th></th>
<th>Ann: -1, Bill: -1</th>
<th>Ann: -4, Bill: 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cooperate</td>
<td>Ann: 0, Bill: -4</td>
<td>Ann: -3, Bill: -3</td>
</tr>
<tr>
<td>Defect</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Defect** is dominant for Ann

**Defect** is dominant for Bill
6.1 Logical Characterisations
Adding preferences

- ATL/CL can express properties about (sequences of) game forms
- In order to reason about solution concepts, we need to add preferences to the picture
- Can be done in several ways
- In lectures 9 and 10, we use preference modalities
- Here we choose a simple solution: “primitive” utility propositions
Utility propositions

Let $U$ be a finite set of utilities. We assume that the primitive propositions $\Pi$ includes a proposition

$$u_i \geq v$$

for each agent $i$ and $v \in U$. 
Utility propositions

Let $U$ be a finite set of utilities. We assume that the primitive propositions $\Pi$ includes a proposition

$$u_i \geq v$$

for each agent $i$ and $v \in U$.

It is now straightforward to identify a strategic game in each state (where the outcomes are new states). We use $\Gamma(M, s)$ to denote the game played in state $s$ of structure $M$. 
Example (BoS)

\[ \Gamma(\mathcal{M}, q_0): \]

<table>
<thead>
<tr>
<th>Ann</th>
<th>B</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ann:2, Bill:1</td>
<td>Ann:0, Bill:0</td>
</tr>
<tr>
<td>B</td>
<td>Ann:0, Bill:0</td>
<td>Ann:1, Bill:2</td>
</tr>
<tr>
<td>S</td>
<td>Ann:0, Bill:0</td>
<td>Ann:1, Bill:2</td>
</tr>
</tbody>
</table>

\[ \mathcal{M}: \]

- \( q_0 \): \( u_A \geq 1, u_B \geq 2 \)
- \( q_1 \): \( u_A \geq 1, u_B \geq 2 \)
- \( q_2 \): \( u_A \geq 0, u_B \geq 0 \)
- \( q_3 \): \( u_A \geq 0, u_B \geq 0 \)
- \( q_4 \): \( u_A \geq 2, u_B \geq 1 \)
We can now express properties of games: $\mathcal{M}, s \models \phi$
means that the game $\Gamma(\mathcal{M}, s)$ has the property described by $\phi$.
We can now express properties of games: $\mathcal{M}, s \models \phi$
means that the game $\Gamma(\mathcal{M}, s)$ has the property described by $\phi$

Game properties of special interest: solution concepts

From now on: simplifying assumption: $N = 2$
We can now express properties of games: \( M, s \models \phi \)
means that the game \( \Gamma(M, s) \) has the property described by \( \phi \).

Game properties of special interest: solution concepts

From now on: simplifying assumption: \( N = 2 \)

Can we express, e.g., Nash equilibrium using the key construct \( \langle C \rangle \)?
We can now express properties of games: $M, s \models \phi$
means that the game $\Gamma(M, s)$ has the property
described by $\phi$

Game properties of special interest: solution concepts

- From now on: simplifying assumption: $N = 2$

Can we express, e.g., Nash equilibrium using the key
construct $\langle C \rangle$?

Turns out to be difficult. Main reasons:

- We can now express properties of games: $\mathcal{M}, s \models \phi$ means that the game $\Gamma(\mathcal{M}, s)$ has the property described by $\phi$.

- Game properties of special interest: solution concepts
  - From now on: simplifying assumption: $N = 2$

- Can we express, e.g., Nash equilibrium using the key construct $\langle \langle C \rangle \rangle$?

- Turns out to be difficult. Main reasons:
  - Solution concepts such as Nash equilibrium are properties of strategies, but we cannot refer directly to strategies in the language.

- We can now express properties of games: \( M, s \models \phi \) means that the game \( \Gamma(M, s) \) has the property described by \( \phi \).
- Game properties of special interest: solution concepts
  - From now on: simplifying assumption: \( N = 2 \)
- Can we express, e.g., Nash equilibrium using the key construct \( \langle \langle C \rangle \rangle \)?
- Turns out to be difficult. Main reasons:
  - Solution concepts such as Nash equilibrium are properties of strategies, but we cannot refer directly to strategies in the language.
  - We often need to reason in the context of a fixed strategy for one or more agents: “If my opponent cooperates, then..”. This requires an “irrevocable” interpretation of strategies.
We can now express properties of games: $\mathcal{M}, s \models \phi$ means that the game $\Gamma(\mathcal{M}, s)$ has the property described by $\phi$.

Game properties of special interest: solution concepts
- From now on: simplifying assumption: $N = 2$

Can we express, e.g., Nash equilibrium using the key construct $\langle \langle C \rangle \rangle$?

Turns out to be difficult. Main reasons:
- Solution concepts such as Nash equilibrium are properties of strategies, but we cannot refer directly to strategies in the language.
- We often need to reason in the context of a fixed strategy for one or more agents: “If my opponent cooperates, then..”. This requires an “irrevocable” interpretation of strategies.
- Reasoning about solution concepts involve counterfactual arguments such as “Suppose my opponent cooperates. Then I better defect. If he defects, however, I should defect as well.”

We will thus make another addition to the language, in addition to utility propositions.
Counterfactuals

- Example: “Suppose my opponent cooperates. Then I better defect. If he defects, however, I should defect as well.”

- Counterfactuals are not logical implications (otherwise one of the claims above would be trivially true)

- Counterfactuals have been analysed by philosophers (Stalnaker, Lewis):
  - “if counterfactually $\phi$ then $\psi$”: if we adjust the world minimally so that $\phi$, then $\psi$
A Counterfactual Operator

Extend the language of ATL (or Coalition Logic) with a counterfactual operator

\[ C_i(\sigma_i, \varphi) \]

where

- \( i \) is an agent
- \( \sigma_i \) is a strategy term. We assume a set of strategy terms \( \Upsilon_i \) for each agent \( i \).
- \( \varphi \) is a formula

with the intended meaning that if \( i \) played strategy \( \sigma_i \), then \( \varphi \) would be true.
A Counterfactual Operator

Extend the language of ATL (or Coalition Logic) with a counterfactual operator

$$C_i(\sigma_i, \varphi)$$

where

- $i$ is an agent
- $\sigma_i$ is a strategy term. We assume a set of strategy terms $\Upsilon_i$ for each agent $i$.
- $\varphi$ is a formula

with the intended meaning that if $i$ played strategy $\sigma_i$, then $\varphi$ would be true

Restriction: no occurrence of a term in $\Upsilon_i$ inside $\varphi$
Interpretation

Extend the semantic structures (CGSs) with an interpretation function $\llbracket \cdot \rrbracket_M$ mapping a strategy term $\sigma_i \in \Upsilon_i$ to a strategy $\llbracket \sigma_i \rrbracket_M$ for agent $i$. 
Interpretation

Extend the semantic structures (CGSs) with an interpretation function $\llbracket \cdot \rrbracket_M$ mapping a strategy term $\sigma_i \in \Upsilon_i$ to a strategy $\llbracket \sigma_i \rrbracket_M$ for agent $i$.

Interpretation:

$\mathcal{M}, q \models C_i(\sigma_i, \varphi) \iff (\mathcal{M} \upharpoonright \llbracket \sigma_i \rrbracket, q \models \varphi)$

1. Logical Characterisations

Interpretation

Extend the semantic structures (CGSs) with an interpretation function \( \llbracket \cdot \rrbracket_M \) mapping a strategy term \( \sigma_i \in \Upsilon_i \) to a strategy \( \llbracket \sigma_i \rrbracket_M \) for agent \( i \).

Interpretation:

\[ M, q \models C_i(\sigma_i, \varphi) \iff (M \upharpoonright \llbracket \sigma_i \rrbracket, q \models \varphi) \]

Assumption: there is a term for every possible strategy.
Note that

\[ \neg \models \langle \langle i \rangle \rangle \varphi \leftrightarrow \bigvee_{\sigma_i \in \Upsilon_i} C_i(\sigma_i, \varphi) \]
Note that

\[ \not\models \langle i \rangle \varphi \leftrightarrow \bigvee_{\sigma_i \in \Upsilon_i} C_i(\sigma_i, \varphi) \]

... similarly to the fact that \( \langle i \rangle \) is different with the standard and the irrevocable semantics: the update semantics rules out any possible future choices.
Characterising Weak Dominance

Find a formula $WD_i(\alpha)$ such that

$$\mathcal{M}, q \models WD_i(\alpha) \iff \alpha \text{ is weakly dominant in } \Gamma(\mathcal{M}, q)$$
Characterising Weak Dominance

Find a formula $WD_i(\alpha)$ such that

$$\mathcal{M}, q \models WD_i(\alpha) \iff \alpha \text{ is weakly dominant in } \Gamma(\mathcal{M}, q)$$

Consider this:

$$wd_i(\alpha) \equiv \bigwedge_{v \in U} (\langle i \rangle \circ (u_i \geq v) \rightarrow C_i(\alpha, \langle \rangle \circ (u_i \geq v)))$$

1. Logical Characterisations

\[ wd_i(\alpha) \equiv \bigwedge_{v \in U} (\langle i \rangle \circ (u_i \geq v) \rightarrow C_i(\alpha, \langle \rangle \circ (u_i \geq v))) \]

1. Logical Characterisations

\[ \text{wd}_i(\alpha) \equiv \bigwedge_{v \in U} (\langle i \rangle \circ (u_i \geq v) \rightarrow C_i(\alpha, \langle \rangle \circ (u_i \geq v))) \]

\[ M: \]

\[ \Gamma(M, q_0): \]

<table>
<thead>
<tr>
<th>Ann</th>
<th>B</th>
<th>S</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Ann:2, Bill:1</td>
<td>Ann:0, Bill: 0</td>
</tr>
<tr>
<td>S</td>
<td>Ann:0, Bill:0</td>
<td>Ann:1, Bill: 2</td>
</tr>
</tbody>
</table>

1. Logical Characterisations

\[ wd_i(\alpha) \equiv \bigwedge_{v \in U} (\langle i \rangle \bigcirc (u_i \geq v) \rightarrow C_i(\alpha, \langle \rangle \bigcirc (u_i \geq v))) \]

\[ M: \]

\[
\begin{array}{c}
q_0 \\
\downarrow \\
S,S \\
q_1 & S,B \\
& q_2 & B,S \\
& & q_3 & B,B \\
& & q_4 \\
& u_A \geq 1, u_B \geq 2 & u_A \geq 0, u_B \geq 0 & u_A \geq 0, u_B \geq 0 & u_A \geq 2, u_B \geq 1
\end{array}
\]

- If \( M, q_0 \models \langle A \rangle \bigcirc u_A \geq v \), then \( v = 0 \)
\[ \text{wd}_i(\alpha) \equiv \bigwedge_{v \in U} (\langle i \rangle \bigcirc (u_i \geq v) \rightarrow C_i(\alpha, \langle \rangle \bigcirc (u_i \geq v))) \]

\[ M \uparrow B_A: \]

- If \( M, q_0 \models \langle A \rangle \bigcirc u_A \geq v \), then \( v = 0 \)
- But \( M \uparrow B_A, q_0 \models \langle \rangle \bigcirc u_A \geq 0 \) (if Ann plays \( B \), she will get at least 0)

1. Logical Characterisations

\[ wd_i(\alpha) \equiv \bigwedge_{v \in U} (\llangle i \rrangle \bigcirc (u_i \geq v) \rightarrow C_i(\alpha, \llangle \rrangle \bigcirc (u_i \geq v))) \]

\[ M: \]

- If \( M, q_0 \models \llangle A \rrangle \bigcirc u_A \geq v \), then \( v = 0 \)
- But \( M \uparrow B_A, q_0 \models \llangle \rrangle \bigcirc u_A \geq 0 \) (if Ann plays \( B \), she will get at least 0)
- So \( M, q_0 \models C_A(B, \llangle \rrangle \bigcirc u_A \geq 0) \) as well

1. Logical Characterisations

\[ wd_i(\alpha) \equiv \bigwedge_{v \in U} (\langle i \rangle \circ (u_i \geq v) \rightarrow C_i(\alpha, \langle \rangle \circ (u_i \geq v))) \]

\[ M: \]

- If \( M, q_0 \models \langle A \rangle \circ u_A \geq v \), then \( v = 0 \)
- But \( M \uparrow B_A, q_0 \models \langle \rangle \circ u_A \geq 0 \) (if Ann plays \( B \), she will get at least 0)
- So \( M, q_0 \models C_A(B, \langle \rangle \circ u_A \geq 0) \) as well
- Thus, \( M, q_0 \models wd_i(B) \)

1. Logical Characterisations

\[ wd_i(\alpha) \equiv \bigwedge_{v \in U} (\langle i \rangle \bigcirc (u_i \geq v) \rightarrow C_i(\alpha, \langle \rangle \bigcirc (u_i \geq v))) \]

\[ M: \]

- If \( M, q_0 \models \langle A \rangle \bigcirc u_A \geq v \), then \( v = 0 \)
- But \( M \uparrow B_A, q_0 \models \langle \rangle \bigcirc u_A \geq 0 \) (if Ann plays \( B \), she will get at least 0)
- So \( M, q_0 \models C_A(B, \langle \rangle \bigcirc u_A \geq 0) \) as well
- Thus, \( M, q_0 \models wd_i(B) \)
- But \( B \) is not a dominant strategy for Ann! (BoS has no dominant strategies)
Characterising Weak Dominance

Find a formula $WD_i(\alpha)$ such that

$$\mathcal{M}, q \models WD_i(\alpha) \iff \alpha \text{ is weakly dominant in } \Gamma(\mathcal{M}, q)$$

$$wd_i(\alpha) \equiv \bigwedge_{v \in U} ((\ll i \gg \odot (u_i \geq v) \rightarrow C_i(\alpha, \ll \gg \odot (u_i \geq v))))$$
Characterising Weak Dominance

Find a formula $WD_i(\alpha)$ such that

$\mathcal{M}, q \models WD_i(\alpha) \iff \alpha$ is weakly dominant in $\Gamma(\mathcal{M}, q)$

$$wd_i(\alpha) \equiv \bigwedge_{v \in U} (\langle i \rangle \bigcirc (u_i \geq v) \rightarrow C_i(\alpha, \langle i \rangle \bigcirc (u_i \geq v)))$$

Solution – for games with finitely many strategies:

$$WD_i(\alpha) \equiv \bigwedge_{\beta \in \mathcal{Y}_j} C_j(\beta, wd_i(\alpha))$$
Characterising Nash Equilibrium

Find a formula $NE_i(\alpha_1, \alpha_2)$ such that $\mathcal{M}, q \models NE(\alpha_1, \alpha_2) \iff (\alpha_1, \alpha_2)$ is a Nash equilibrium of $\Gamma(\mathcal{M}, q)$
Characterising Nash Equilibrium

Find a formula $NE_i(\alpha_1, \alpha_2)$ such that

$\mathcal{M}, q \models NE(\alpha_1, \alpha_2) \iff (\alpha_1, \alpha_2)$ is a Nash equilibrium of $\Gamma(\mathcal{M}, q)$

Best response:

$BR_i(\alpha_k, \alpha_i) \equiv C_k(\alpha_k, \bigwedge_{v \in U} (((\langle i \rangle \bigcirc (u_i \geq v)) \rightarrow C_i(\alpha_i, \langle \rangle \bigcirc (u_i \geq v))))$
Characterising Nash Equilibrium

Find a formula $\text{NE}_i(\alpha_1, \alpha_2)$ such that

$\mathcal{M}, q \models \text{NE}(\alpha_1, \alpha_2) \iff (\alpha_1, \alpha_2)$ is a Nash equilibrium of $\Gamma(\mathcal{M}, q)$

Best response:

$\text{BR}_i(\alpha_k, \alpha_i) \equiv C_k(\alpha_k, \bigwedge_{v \in U} (((i) \circ (u_i \geq v)) \rightarrow C_i(\alpha_i, (i) \circ (u_i \geq v))))$

$\text{NE}(\alpha_1, \alpha_2) \equiv \text{BR}_1(\alpha_2, \alpha_1) \land \text{BR}_2(\alpha_1, \alpha_2)$
6.2 References

2. References

A Course in Game Theory.  

A logic for strategic reasoning.  

Modal logic and game theory: Two alternative approaches.  

A modal characterization of Nash equilibrium.  
7. Reasoning about Rational Play

Reasoning about Rational Play
Game-Theoretical Analysis of Games

- **Solution concepts** define rationality of players

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.
7. Reasoning about Rational Play

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
7. Reasoning about Rational Play

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
7. Reasoning about Rational Play

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
Example: Pennies Game
Example: Pennies Game

\[
\text{start} \rightarrow \neg \langle 1 \rangle \Diamond \text{money}_1
\]
7. Reasoning about Rational Play

Example: Pennies Game

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

\[
\text{start} \rightarrow \neg \langle 2 \rangle \Diamond \text{money}_2
\]
Example: Pennies Game

start $\rightarrow \neg \langle 1 \rangle \Diamond \text{money}_1$

start $\rightarrow \neg \langle 2 \rangle \Diamond \text{money}_2$
7. Reasoning about Rational Play

Game-Theoretical Analysis of Games

Two points of focus:

- **characterization** of rationality
  
  \[ \leadsto \]  research in game theory
7. Reasoning about Rational Play

Game-Theoretical Analysis of Games

Two points of focus:

- **characterization** of rationality
  \[\rightsquigarrow\] research in game theory

- **using** solution concepts to predict outcomes in a given game
  \[\rightsquigarrow\] applications of game theory
Motivation

We would like to ...

... reason about the outcome of rational play
Motivation

We would like to ...

... reason about the outcome of rational play
... have a logic that embed any solution concept
... compare different game theoretical solution concepts wrt their outcomes
Motivation

We would like to ...

... reason about the outcome of rational play
... have a logic that embed any solution concept
... compare different game theoretical solution concepts wrt their outcomes

So ...

... we extend ATL with a notion of rationality/plausibility
... reason about what rational agents can achieve
7. Reasoning about Rational Play

Inspiration:

- Game Logics with Preferences (van Otterloo, van der Hoek & Wooldridge): Nash equilibria, subgame perfect strategies
- Epistemic Temporal Strategic Logic (van Otterloo & Jonker): undominated strategies
7. Reasoning about Rational Play

1. ATL + Plausibility

7.1 ATL + Plausibility
ATL with Plausibility

ATL: reasoning about all possible behaviors.

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

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

\[
\langle \langle A \rangle \rangle \varphi : \text{agents } A \text{ have *some* collective strategy to enforce } \varphi \text{ against *any* response of their opponents.}
\]

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

\[
\mathcal{P} \langle \langle A \rangle \rangle \varphi : \text{agents } A \text{ have a *plausible* collective strategy to enforce } \varphi \text{ against any *plausible* response of their opponents.}
\]
ATL with Plausibility

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

\[\langle A \rangle \varphi: \text{agents } A \text{ have some collective strategy to enforce } \varphi \text{ against any response of their opponents.}\]

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

\[\text{Pl } \langle A \rangle \varphi: \text{agents } A \text{ have a plausible collective strategy to enforce } \varphi \text{ against any plausible response of their opponents.}\]

**Important**

The possible strategies of both \(A\) and \(\text{Agt} \setminus A\) are restricted.
ATL with Plausibility

Syntax of ATLP

\[ \varphi ::= p | \neg \varphi | \varphi \land \varphi | \langle A \rangle \lozenge \varphi | \langle A \rangle \square \varphi | \langle A \rangle \varphi U \varphi | (\text{set-pl } \omega) \varphi \]
ATL with Plausibility

Syntax of ATLP

\[ \varphi ::= p \mid \neg \varphi \mid \varphi \land \varphi \mid \langle A \rangle \lozenge \varphi \mid \langle A \rangle \Box \varphi \mid \langle A \rangle \varphi U \psi \mid (\text{set-pl } \omega) \varphi \]

New in ATLP:

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

Syntax of ATLP

\[ \varphi ::= p \mid \neg \varphi \mid \varphi \land \varphi \mid \langle A \rangle \Diamond \varphi \mid \langle A \rangle \Box \varphi \mid \langle A \rangle \varphi \mathcal{U} \varphi \mid (\text{set-pl} \ \omega) \varphi \]

New in ATLP:

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

Example: \( (\text{set-pl} \ greedy_1) \langle 2 \rangle \Diamond \text{money}_2 \)
Concurrent Game Structures with Plausibility

\[ M = (\text{Agt}, \text{St}, \Pi, \pi, \text{Act}, d, \delta, \Upsilon, \Omega, \parallel \cdot \parallel) \]
Concurrent Game Structures with Plausibility

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

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

\[
\begin{array}{c|cc}
 & \text{Deny} & \text{Confess} \\
\hline
\text{Deny} & -2, -2 & -5, -1 \\
\text{Confess} & -1, -5 & -4, -4 \\
\end{array}
\]
Concurrent Game Structures with Plausibility

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

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

\[ \Omega = \{ \omega_1, \omega_2, \ldots \} \]: set of plausibility terms

Example: \( \omega_{NE} \) may stand for all Nash equilibria
Concurrent Game Structures with Plausibility

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

- \[ \Upsilon \subseteq \Sigma \]: set of plausible strategy profiles

- \[ \Omega = \{\omega_1, \omega_2, \ldots\} \]: set of plausibility terms

Example: \( \omega_{NE} \) may stand for all Nash equilibria

- \[ \| \cdot \| : St \to (\Omega \to 2^\Sigma) \]: plausibility mapping, assigns set of strategy profiles to each state and plausibility term

Example: \( \| \omega_{NE} \|_q = \{(\text{confess, confess})\} \)
Semantics of ATLP

$\Sigma_A(\Upsilon)$: collective strategies of $A$ that are consistent with $\Upsilon$

Restricting $A$’s strategies

$\Sigma_A(\Upsilon) = \{ s_A \in \Sigma_A | \exists t \in \Upsilon \ (t[A] = s_A) \}$
Semantics of ATLP

\[ \Sigma_A(\Upsilon): \text{collective strategies of } A \text{ that are consistent with } \Upsilon \]

Restricting A’s strategies

\[ \Sigma_A(\Upsilon) = \{ s_A \in \Sigma_A \mid \exists t \in \Upsilon \ (t[A] = s_A) \} \]

We also restrict the opponents’ responses to \( s_A \)

\[ \Upsilon(s_A): \text{plausible strategy profiles of } \text{Aggt} \text{ that agree on } s_A \]

Restricting A’s opponents strategies

\[ \Upsilon(s_A) = \{ t \in \Upsilon \mid t[A] = s_A \} \]
7. Reasoning about Rational Play

1. ATL + Plausibility

Restricting Strategies

\[ \Upsilon = \{(\text{confess}_1, \text{confess}_2), (\text{deny}_1, \text{deny}_2)\} \]

\[
\begin{array}{c|cc}
 & \text{Deny} & \text{Confess} \\
\hline
\text{Deny} & -2, -2 & -5, -1 \\
\text{Confess} & -1, -5 & -4, -4 \\
\end{array}
\]
Restricting Strategies

\[ \Upsilon = \{ (\text{confess}_1, \text{confess}_2), (\text{deny}_1, \text{deny}_2) \} \]

\[ \Sigma_1(\Upsilon) = \{ \text{confess}_1, \text{deny}_1 \} \]
Restricting Strategies

\[ \Upsilon = \{ (\text{confess}_1, \text{confess}_2), (\text{deny}_1, \text{deny}_2) \} \]
\[ \Sigma_1(\Upsilon) = \{ \text{confess}_1, \text{deny}_1 \} \]
\[ P(\text{confess}_1) = \{ (\text{confess}_1, \text{confess}_2) \} \]
7. Reasoning about Rational Play

1. ATL + Plausibility

Outcome of a Strategy

Outcome = Paths that may occur when agents $A$ perform $S_A$
Outcome of a Strategy

Outcome = Paths that may occur when agents $A$ perform $s_A$ and only plausible strategy profiles are played
Outcome of a Strategy

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

$$\text{outcome}(q, s_A) =$$

$$\{ \lambda \in St^+ | \exists t \in \Upsilon(s_A) \forall i \in \mathbb{N} \ (\lambda[i + 1] = \delta(\lambda[i], t(\lambda[i]))) \}$$
Outcomes of a Strategy

Outcomes = Paths that may occur when agents $A$ perform $s_A$ and only plausible strategy profiles are played

$$\text{out}_{\Upsilon}(q, s_A) = \{ \lambda \in St^+ | \exists t \in \Upsilon(s_A) \forall i \in \mathbb{N} (\lambda[i + 1] = \delta(\lambda[i], t(\lambda[i]))) \}$$
7. Reasoning about Rational Play

1. ATL + Plausibility

Outcome of a Strategy

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

\[ \text{out}_{\Upsilon}(q, s_A) = \{ \lambda \in St^+ | \exists t \in \Upsilon(s_A) \forall i \in \mathbb{N} \left( \lambda[i + 1] = \delta(\lambda[i], t(\lambda[i])) \right) \} \]

\[ q_0 \rightarrow q_1 \rightarrow q_2 \rightarrow q_3 \rightarrow q_4 \rightarrow q_5 \]

$P$: the players always show same sides of their coins
7. Reasoning about Rational Play

1. ATL + Plausibility

Outcome of a Strategy

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

\[\text{outcome}_{\Upsilon}(q, s_A) = \{ \lambda \in St^+ | \exists t \in \Upsilon(s_A) \forall i \in \mathbb{N} \ (\lambda[i + 1] = \delta(\lambda[i], t(\lambda[i]))) \}\]

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

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

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

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

$$M, q_0 \models (\text{set-pl sameside}) \langle \emptyset \rangle \lozenge \text{money}_1$$
Example: Pennies Game

\[ M, q_0 \models (\text{set-pl sameside}) \langle \emptyset \rangle \bigcirc \text{money}_1 \]
Example: Pennies Game

\[ M, q_0 \models (\text{set-pl sameside}) \langle \emptyset \rangle \diamond \text{money}_1 \]

\[ M, q_0 \models (\text{set-pl } \omega_{NE}) \langle 2 \rangle \Box \text{money}_2 \]
7. Reasoning about Rational Play

1. ATL + Plausibility

Example: Pennies Game

What is a Nash equilibrium in this game?
We need some kind of winning criteria!

\[ M, q_0 \models (\text{set-pl sameside}) \langle \emptyset \rangle \lozenge \text{money}_1 \]

\[ M, q_0 \models (\text{set-pl } \omega_{NE}) \langle 2 \rangle \boxdot \text{money}_2 \]
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.
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.
7. Reasoning about Rational Play

1. ATL + Plausibility

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.
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.
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.

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

where $\| \omega_{NE} \|_{q_0} = “\text{all profiles belonging to grey cells}”$. 
What about games with non-binary payoffs?
What about games with non-binary payoffs?

- Option 1: instead of a single winning condition, we use a list of conditions to encode preferences over outcomes
What about games with non-binary payoffs?

- Option 1: instead of a single winning condition, we use a list of conditions to encode preferences over outcomes.
- Option 2: we use the construction by Baltag to embed utilities in CGS, and then refer to temporal patterns of utilities.
What about games with non-binary payoffs?

- Option 1: instead of a single winning condition, we use a list of conditions to encode preferences over outcomes.

- Option 2: we use the construction by Baltag to embed utilities in CGS, and then refer to temporal patterns of utilities.

- Simplest characteristic of such patterns: the utility obtained eventually at the end of the game.
7. Reasoning about Rational Play

1. ATL + Plausibility

Extensive Games as Concurrent Game Structures
The Construction

- Model terminal nodes as "sink" states
- Emulate utilities with propositions
- \( M, q \models u_a \geq v \): "\( a \) gets at least \( v \) in state \( q \)"
The Construction

- Model terminal nodes as “sink” states
- Emulate utilities with propositions
- $M, q \models u_a \geq v$: “$a$ gets at least $v$ in state $q$”

Now: CGS are a generalization of extensive games
7. Reasoning about Rational Play

1. ATL + Plausibility

Extensive Games as Concurrent Game Structures
7. Reasoning about Rational Play

1. ATL + Plausibility

Temporalized Solution Concepts

- Outcome of a game:
  - in an extensive game: single utility value
  - in CGS: infinite temporal path
Temporalized Solution Concepts

- Outcome of a game:
  - in an extensive game: single utility value
  - in CGS: infinite temporal path

\[ \rightsquigarrow \text{temporal pattern of utilities} \]
Temporalized Solution Concepts

- Outcome of a game:
  - in an extensive game: single utility value
  - in CGS: infinite temporal path
  \[ \Rightarrow \text{temporal pattern of utilities} \]
- (In extensive games, paths are identical to states – in CGS not!)
Temporalized Solution Concepts

- Outcome of a game:
  - in an extensive game: single utility value
  - in CGS: infinite temporal path
  - $\leadsto$ temporal pattern of utilities

- (In extensive games, paths are identical to states – in CGS not!)

- We need to define the payoff for agent $a$ of path $\lambda$

- Qualitative approach: see previous slides

- Quantitative approach: guaranteed utility ($\leadsto a$ gets always at least $u$), achievable utility ($\leadsto a$ gets eventually at least $u$)...?

- $\square u_a \geq 1$, $\Diamond u_a \geq 1$, $\bigcirc u_a \geq 1$, ...
Temporalized Solution Concepts

- Outcome of a game:
  - in an extensive game: single utility value
  - in CGS: infinite temporal path
  \[ \rightsquigarrow \text{temporal pattern of utilities} \]

- (In extensive games, paths are identical to states – in CGS not!)

- We need to define the payoff for agent \( a \) of path \( \lambda \)
- Qualitative approach: see previous slides
- Quantitative approach: guaranteed utility (\( \rightsquigarrow a \) gets always at least \( u \)), achievable utility (\( \rightsquigarrow a \) gets eventually at least \( u \))...
- \( \Box u_a \geq 1, \Diamond u_a \geq 1, \bigcirc u_a \geq 1, ... \)
- ...Temporalized solution concepts (parameterized with temporal operators)
Temporalized Solution Concepts

\[ M, q_0 \models (\text{set-pl } \omega_{NE}) \langle 2 \rangle \Diamond (u_2 \geq 1) \]
7.2 Plausibility Specifications
How to Obtain Plausibility Terms?

Plausibility terms: *abstract labels, no structure!*
How to Obtain Plausibility Terms?

Plausibility terms: abstract labels, no structure!

Idea

Formulae that describe plausible strategies!

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

We need to “plug in” logical characterizations of rationality assumptions ⇝ CATL
But: in fact, we can use ATLP instead! ⇝

The same language for characterizing rationality and reasoning about the outcome of rational play
How to Obtain Plausibility Terms?

Plausibility terms: abstract labels, no structure!

Idea

Formulae that describe plausible strategies!

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

We need to “plug in” logical characterizations of rationality assumptions $\leadsto$ CATL
How to Obtain Plausibility Terms?

Plausibility terms: abstract labels, no structure!

Idea

Formulae that describe plausible strategies!

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

We need to “plug in” logical characterizations of rationality assumptions \(\leadsto\) CATL

But: in fact, we can use ATLP instead!
How to Obtain Plausibility Terms?

Plausibility terms: abstract labels, no structure!

**Idea**

Formulae that describe plausible strategies!

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

We need to “plug in” logical characterizations of rationality assumptions $\leadsto$ CATL

But: in fact, we can use ATLP instead!  
$\leadsto$ The same language for characterizing rationality and reasoning about the outcome of rational play
Sometimes quantifiers are needed...

E.g.: $\text{set-pl } \sigma. \forall \sigma' \text{ dominates} (\sigma, \sigma')$
ATLP: Extending the Syntax

Definition 7.1 (Logics $\mathcal{L}^k_{\text{ATLP}}$)

Let $\Omega$ be a set of primitive plausibility terms, and $\mathcal{V}ar$ be a set of strategic variables (with typical element $\omega$). $\mathcal{L}^k_{\text{ATLP}}(\text{Agt}, \Pi, \mathcal{V}ar, \Omega)$ are defined recursively:

- $\mathcal{L}^0_{\text{ATLP}}(\text{Agt}, \Pi, \mathcal{V}ar, \Omega) = \mathcal{L}^\text{base}_{\text{ATLP}}(\text{Agt}, \Pi, \Omega_0)$
  where $\Omega_0 = T(\Omega)$;

- $\mathcal{L}^k_{\text{ATLP}}(\text{Agt}, \Pi, \mathcal{V}ar, \Omega) = \mathcal{L}^\text{base}_{\text{ATLP}}(\text{Agt}, \Pi, \Omega_k)$, where:
  - $\Omega_k := T(\Omega_{k-1} \cup \Omega^k)$,
  - $\Omega^k := \{\sigma_1.(Q_2\sigma_2)\ldots(Q_n\sigma_n)\varphi | n \in \mathbb{N}, \forall i \ (1 \leq i \leq n \Rightarrow \sigma_i \in \mathcal{V}ar, \ Q_i \in \{\forall, \exists\}, \ \varphi \in \mathcal{L}^\text{base}_{\text{ATLP}}(\text{Agt}, \Pi, T(\Omega_{k-1} \cup \{\sigma_1, \ldots, \sigma_n\}))) \}$. 
ATLP: Extending the Syntax

Definition 7.1 (Logics $\mathcal{L}^k_{ATLP}$)

Let $\Omega$ be a set of primitive plausibility terms, and $Var$ be a set of strategic variables (with typical element $\omega$). $\mathcal{L}^k_{ATLP}(\text{Agt}, \Pi, Var, \Omega)$ are defined recursively:

- $\mathcal{L}^0_{ATLP}(\text{Agt}, \Pi, Var, \Omega) = \mathcal{L}^{\text{base}}_{ATLP}(\text{Agt}, \Pi, \Omega_0)$ where $\Omega_0 = \mathcal{T}(\Omega)$;
- $\mathcal{L}^k_{ATLP}(\text{Agt}, \Pi, Var, \Omega) = \mathcal{L}^{\text{base}}_{ATLP}(\text{Agt}, \Pi, \Omega_k)$, where:
  - $\Omega_k := \mathcal{T}(\Omega_{k-1} \cup \Omega^k)$,
  - $\Omega^k := \{\sigma_1.(Q_2\sigma_2)\cdots(Q_n\sigma_n)\varphi \mid n \in \mathbb{N}, \forall i (1 \leq i \leq n \Rightarrow \sigma_i \in Var, Q_i \in \{\forall, \exists\}, \varphi \in \mathcal{L}^{\text{base}}_{ATLP}(\text{Agt}, \Pi, \mathcal{T}(\Omega_{k-1} \cup \{\sigma_1, \ldots, \sigma_n\}))) \}$.

The set of ATLP formulae with arbitrary finite nesting of plausibility terms is defined by $\mathcal{L}^\infty_{ATLP}$.
Formal Semantics...
Formal Semantics...
Formal Semantics... let’s jump over it
7.3 Characterizations
Qualitative Characterization of Nash Equilibrium

\( \sigma_a \) is \( a \)'s best response to \( \sigma \) (wrt \( \overline{\gamma} \)):

\[
BR_{\overline{\gamma}}(\sigma) \equiv (\text{set-pl } \sigma[\text{Agt} \setminus \{a\}])(\langle a \rangle \gamma_a \rightarrow (\text{set-pl } \sigma)\langle \emptyset \rangle \gamma_a)
\]
Qualitative Characterization of Nash Equilibrium

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

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

$\sigma$ is a Nash equilibrium:

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

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

\[ \begin{array}{c|cccc}
\gamma_1 \setminus \gamma_2 & hh & ht & th & tt \\
\hline
HH & 1, 1 & 0, 0 & 0, 1 & 0, 1 \\
HT & 0, 0 & 0, 1 & 0, 1 & 0, 1 \\
TH & 0, 1 & 0, 1 & 1, 1 & 0, 0 \\
TT & 0, 1 & 0, 1 & 0, 0 & 0, 1 \\
\end{array} \]

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

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

\( \sigma \) is a subgame perfect Nash equilibrium:

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

\( \sigma \) is Pareto optimal:

\[
PO^\gamma (\sigma) \equiv \forall \sigma' \left( \bigwedge_{a \in \text{Agt}} ((\text{set-pl } \sigma') \langle \emptyset \rangle \gamma_a \rightarrow (\text{set-pl } \sigma) \langle \emptyset \rangle \gamma_a) \lor \bigvee_{a \in \text{Agt}} ((\text{set-pl } \sigma) \langle \emptyset \rangle \gamma_a \land \neg (\text{set-pl } \sigma') \langle \emptyset \rangle \gamma_a) \right).
\]
Characterizations of Other Solution Concepts

\( \sigma \) is **undominated**: 

\[
\text{UNDOM}^\gamma(\sigma) \equiv \forall \sigma_1 \forall \sigma_2 \exists \sigma_3 \\
\left( ((\text{set-pl} \ \langle \sigma_1 \{a\}, \sigma_2^{\text{Agt}\backslash\{a\}} \rangle) \langle\emptyset\rangle \gamma_a \rightarrow \\
(\text{set-pl} \ \langle \sigma_1 \{a\}, \sigma_2^{\text{Agt}\backslash\{a\}} \rangle) \langle\emptyset\rangle \gamma_a) \\
\lor ((\text{set-pl} \ \langle \sigma_1 \{a\}, \sigma_3^{\text{Agt}\backslash\{a\}} \rangle) \langle\emptyset\rangle \gamma_a \land \\
\neg((\text{set-pl} \ \langle \sigma_1 \{a\}, \sigma_3^{\text{Agt}\backslash\{a\}} \rangle) \langle\emptyset\rangle \gamma_a)) \right).
\]
Example: Pennies Game again

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

\[ M, q_0 \models (\text{set-pl } \sigma. PO^{\gamma_1, \gamma_2}(\sigma) \langle \emptyset \rangle) \Diamond (\text{money}_1 \land \text{money}_2) \]
Theorem 7.2

Let $M$ be a CGSP, $q$ a state in $M$, and $\vec{\eta} = \langle \eta_1, \ldots, \eta_k \rangle$ a vector of path formulae (winning conditions). Moreover, let $\Gamma(M, q, \vec{\eta})$ be the strategic game obtained from $M, q$ by assigning strategy profiles with binary payoffs according to $\vec{\eta}$. Then the following holds:

1. $\| \sigma.NE^\eta(\sigma) \|_{M,q}$ denotes the set of Nash equilibria in $\Gamma(M, q, \vec{\eta})$;
2. $\| \sigma.PO^\eta(\sigma) \|_{M,q}$ denotes the set of Pareto optimal strategy profiles in $\Gamma(M, q, \vec{\eta})$;
3. $\| \sigma.UNDOM^\eta(\sigma) \|_{M,q}$ denotes the set of undominated strategies in $\Gamma(M, q, \vec{\eta})$;
4. $\| \sigma.SPN^\eta(\sigma) \|_{M,q}$ denotes the set of strategy profiles that are in Nash equilibrium for every $\Gamma(M, q', \vec{\eta})$ (for all reachable $q'$).
Temporalized Solution Concepts

Nash Equilibrium:

\[ BR^T_a(\sigma) \equiv (\text{str}_{\text{Agt}\setminus A} \sigma[\text{Agt}\setminus \{a\}]) \]

\[ (\bigwedge_{v \in U} (\langle a \rangle T(u_a \geq v)) \rightarrow (\text{str}_a \sigma[a]) \langle \emptyset \rangle T(u_a \geq v)) \]
Temporalized Solution Concepts

Nash Equilibrium:

\[ BR^T_a(\sigma) \equiv (\text{str}_{\text{Agt}\setminus \{a\}} \sigma[\text{Agt}\setminus \{a\}]) \]

\[ (\bigwedge_{v \in U} (\langle \langle a \rangle \rangle T(u_a \geq v)) \rightarrow (\text{str}_a \sigma[a])\langle \emptyset \rangle T(u_a \geq v)) \]

\[ NE^T(\sigma) \equiv \bigwedge_{a \in \text{Agt}} BR^T_a(\sigma) \]
Temporalized Solution Concepts

Nash Equilibrium:

\[ BR^T_a(\sigma) \equiv (\text{str}_{\text{Agt}\setminus A} \sigma[A\setminus \{a\}]) \]
\[ (\bigwedge_{v \in U} (\langle a \rangle T(u_a \geq v)) \rightarrow (\text{str}_a \sigma[a])\langle\emptyset\rangle T(u_a \geq v)) \]

\[ NE^T(\sigma) \equiv \bigwedge_{a \in \text{Agt}} BR^T_a(\sigma) \]

\[ SPN^T(\sigma) \equiv \langle\emptyset\rangle \square NE^T(\sigma) \]
Temporalized Solution Concepts

\[ PO^T(\sigma) \equiv \forall \sigma' \left( \bigwedge_{a \in \text{Agt}} \bigwedge_{v \in U} \left( (\text{set-pl } \sigma') \text{Pl } \langle \emptyset \rangle T(u_a \geq v) \rightarrow \right. \right. \]
\[
\left. \left. \left( \text{set-pl } \sigma \right) \text{Pl } \langle \emptyset \rangle T(u_a \geq v) \right) \lor \right. \right. \]
\[
\left. \left. \bigvee_{a \in \text{Agt}} \bigvee_{v \in U} \left( (\text{set-pl } \sigma) \text{Pl } \langle \emptyset \rangle T(u_a \geq v) \right) \land \right. \right. \]
\[
\left. \left. \left. \neg (\text{set-pl } \sigma') \text{Pl } \langle \emptyset \rangle T(u_a \geq v) \right) \right) \right). \]
Temporized Solution Concepts

\[ \text{UNDOM}^T(\sigma) \equiv \forall \sigma_1 \forall \sigma_2 \exists \sigma_3 \]
\[
\left( \bigwedge_{v \in U} \left( \text{set-pl} \langle \sigma_1 \{a\}, \sigma_2 \text{\textlogical\{\text{Agent}\}\{\{a\}\} \rangle} \text{Pl} \langle \emptyset \rangle T(u_a \geq v) \rightarrow \right.ight.
\[
\left( \text{set-pl} \langle \sigma \{a\}, \sigma_2 \text{\textlogical\{\text{Agent}\}\{\{a\}\} \rangle} \text{Pl} \langle \emptyset \rangle T(u_a \geq v) \right)
\right.
\]
\[
\left. \bigvee_{v \in U} \left( \text{set-pl} \langle \sigma \{a\}, \sigma_3 \text{\textlogical\{\text{Agent}\}\{\{a\}\} \rangle} \text{Pl} \langle \emptyset \rangle T(u_a \geq v) \right) \land \right.
\]
\[
\neg \left( \text{set-pl} \langle \sigma_1 \{a\}, \sigma_3 \text{\textlogical\{\text{Agent}\}\{\{a\}\} \rangle} \text{Pl} \langle \emptyset \rangle T(u_a \geq v) \right) \right).\]
7. Reasoning about Rational Play

3. Characterizations

Temporalized Solution Concepts

**Theorem 7.3**

Let $\Gamma$ be an extensive game with a finite set of utilities. Then the following holds:

1. $s \in \| \sigma.NE^\Diamond(\sigma) \|_{M(\Gamma),\emptyset}$ iff $s$ is a Nash equilibrium in $\Gamma$;
2. $s \in \| \sigma.SPN^\Diamond(\sigma) \|_{M(\Gamma),\emptyset}$ iff $s$ is a subgame perfect Nash equilibrium in $\Gamma$;
3. $s \in \| \sigma.PO^\Diamond(\sigma) \|_{M(\Gamma),\emptyset}$ iff $s$ is Pareto optimal in $\Gamma$;
4. $s \in \| \sigma.UNDOM^\Diamond(\sigma) \|_{M(\Gamma),\emptyset}$ iff $s$ is undominated in $\Gamma$. 
7.4 Model Checking
Solving Games through Model Checking ATLP

- Concurrent game structure = generalized extensive game
- Plausibility specification $\rightsquigarrow$ solution concept
Solving Games through Model Checking ATLP

- Concurrent game structure = generalized extensive game
- Plausibility specification \( \rightsquigarrow \) solution concept
- \( \langle A \rangle \gamma \) defines a game where \( A \) want to achieve \( \gamma \)
Solving Games through Model Checking ATLP

- Concurrent game structure = generalized extensive game
- Plausibility specification $\leadsto$ solution concept
- $\langle A \rangle \gamma$ defines a game where $A$ want to achieve $\gamma$
  2-player, binary, zero-sum game
Solving Games through Model Checking ATLP

- Concurrent game structure = generalized extensive game
- Plausibility specification $\rightsquigarrow$ solution concept
- $\langle A \rangle \gamma$ defines a game where $A$ want to achieve $\gamma$
  2-player, binary, zero-sum game
  $\rightsquigarrow$ players, payoffs
Solving Games through Model Checking ATLP

- Concurrent game structure = generalized extensive game
- Plausibility specification $\rightsquigarrow$ solution concept
- $\langle A \rangle \gamma$ defines a game where $A$ want to achieve $\gamma$
  2-player, binary, zero-sum game
  $\rightsquigarrow$ players, payoffs
- Model checking formulae of ATLP $\rightsquigarrow$ solving games
# Model checking complexity of ATLP

<table>
<thead>
<tr>
<th>$L^i_{ATLP}$</th>
<th>0</th>
<th>1</th>
<th>...</th>
<th>$i$</th>
<th>...</th>
<th>$\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L^\text{basic}_{ATLP}$</td>
<td>$P$</td>
<td>-</td>
<td>...</td>
<td>-</td>
<td>...</td>
<td>-</td>
</tr>
<tr>
<td>$L^0_{ATLP}$</td>
<td>$P$</td>
<td>-</td>
<td>...</td>
<td>-</td>
<td>...</td>
<td>-</td>
</tr>
<tr>
<td>$L^1_{ATLP}$</td>
<td>$\Delta^P_3$</td>
<td>$\Delta^P_4$</td>
<td>...</td>
<td>$\Delta^P_{i+3}$</td>
<td>...</td>
<td>PSPACE</td>
</tr>
<tr>
<td>$L^2_{ATLP}$</td>
<td>$\Delta^P_4$</td>
<td>$\Delta^P_6$</td>
<td>...</td>
<td>$\Delta^P_{5+i-\max{0,1-i}}$</td>
<td>...</td>
<td>PSPACE</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$L^k_{ATLP}$</td>
<td>$\Delta^P_{k+2}$</td>
<td>$\Delta^P_{k+4}$</td>
<td>...</td>
<td>$\Delta^P_{i+2k+1-\max{0,k-i-1}}$</td>
<td>...</td>
<td>PSPACE</td>
</tr>
</tbody>
</table>

**SAT/mechanism design complexity:** open!
Model checking complexity of ATLP

<table>
<thead>
<tr>
<th>( \mathcal{L}_{ATLP} )</th>
<th>0</th>
<th>1</th>
<th>\ldots</th>
<th>( i )</th>
<th>\ldots</th>
<th>( \infty )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathcal{L}_{ATLP}^{\text{basic}} )</td>
<td>( P )</td>
<td>-</td>
<td>\ldots</td>
<td>-</td>
<td>\ldots</td>
<td>-</td>
</tr>
<tr>
<td>( \mathcal{L}_{ATLP}^{0} )</td>
<td>( P )</td>
<td>-</td>
<td>\ldots</td>
<td>-</td>
<td>\ldots</td>
<td>-</td>
</tr>
<tr>
<td>( \mathcal{L}_{ATLP}^{1} )</td>
<td>( \Delta_{3}^{P} )</td>
<td>( \Delta_{4}^{P} )</td>
<td>\ldots</td>
<td>( \Delta_{i+3}^{P} )</td>
<td>\ldots</td>
<td>\text{PSPACE}</td>
</tr>
<tr>
<td>( \mathcal{L}_{ATLP}^{2} )</td>
<td>( \Delta_{4}^{P} )</td>
<td>( \Delta_{6}^{P} )</td>
<td>\ldots</td>
<td>( \Delta_{5+i-\max{0,1-i}}^{P} )</td>
<td>\ldots</td>
<td>\text{PSPACE}</td>
</tr>
<tr>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
</tr>
<tr>
<td>( \mathcal{L}_{ATLP}^{k} )</td>
<td>( \Delta_{k+2}^{P} )</td>
<td>( \Delta_{k+4}^{P} )</td>
<td>\ldots</td>
<td>( \Delta_{i+2k+1-\max{0,k-i-1}}^{P} )</td>
<td>\ldots</td>
<td>\text{PSPACE}</td>
</tr>
</tbody>
</table>

SAT/mechanism design complexity: open!

8. Model checking

Model checking
8.1 Model Checking Time and Strategies
Model Checking

Model checking: **Does** \( \varphi \) **hold in model** \( \mathcal{M} \) **and state** \( q \)?
Model Checking

Model checking: Does $\varphi$ hold in model $\mathcal{M}$ and state $q$?

Two perspectives to model checking MAS:
Model Checking

Model checking: Does $\varphi$ hold in model $\mathcal{M}$ and state $q$?

Two perspectives to model checking MAS:

- Model represents the view of an objective observer
- Formula: specification to be met
Model Checking

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

Two perspectives to model checking MAS:

**Verification**
- Model represents the view of an *objective observer*
- Formula: *specification* to be met
Model Checking

Model checking: \( \varphi \) hold in model \( M \) and state \( q \)?

Two perspectives to model checking MAS:

**Verification**
- Model represents the view of an *objective observer*
- Formula: *specification* to be met

**Planning**
- Model represents the *subjective* view of an *agent*
- Formula: *goal* to be achieved
Model Checking

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

Two perspectives to model checking MAS:

Verification
- Model represents the view of an objective observer
- Formula: specification to be met

Planning
- Model represents the subjective view of an agent
- Formula: goal to be achieved
**function** \( mcheck(M, \varphi) \).

Model checking formulae of ATL.

Returns the exact subset of \( St \) for which formula \( \varphi \) holds.

<table>
<thead>
<tr>
<th>Case</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \varphi \equiv p )</td>
<td>return ( { q \in St \mid p \in \pi(q) } )</td>
</tr>
<tr>
<td>( \varphi \equiv \neg \psi )</td>
<td>return ( St \setminus mcheck(M, \psi) )</td>
</tr>
<tr>
<td>( \varphi \equiv \psi_1 \land \psi_2 )</td>
<td>return ( mcheck(M, \psi_1) \cap mcheck(M, \psi_2) )</td>
</tr>
<tr>
<td>( \varphi \equiv \langle A \rangle \bigcirc \psi )</td>
<td>return ( pre(A, mcheck(M, \psi)) )</td>
</tr>
<tr>
<td>( \varphi \equiv \langle A \rangle \Box \psi )</td>
<td>( Q_1 := Q; \ Q_2 := Q_3 := mcheck(M, \psi); )</td>
</tr>
<tr>
<td></td>
<td>( \text{while } Q_1 \not\subseteq Q_2 \text{ do } Q_1 := Q_1 \cap Q_2; \ Q_2 := pre(A, Q_1) \cap Q_3 \text{ od; } )</td>
</tr>
<tr>
<td></td>
<td>return ( Q_1 )</td>
</tr>
<tr>
<td>( \varphi \equiv \langle A \rangle \psi_1 \bigcup \psi_2 )</td>
<td>( Q_1 := \emptyset; \ Q_2 := mcheck(M, \psi_2); \ Q_3 := mcheck(M, \psi_1); )</td>
</tr>
<tr>
<td></td>
<td>( \text{while } Q_2 \not\subseteq Q_1 \text{ do } Q_1 := Q_1 \cup Q_2; \ Q_2 := pre(A, Q_1) \cap Q_3 \text{ od; } )</td>
</tr>
<tr>
<td></td>
<td>return ( Q_1 )</td>
</tr>
</tbody>
</table>
Example: Simple Rocket Domain
Example: Simple Rocket Domain

- Verification example: we want to make sure that agents 1 and 3 can move the cargo to the other location.
- caL → ⟨⟨1, 3⟩⟩♦caP
Example: Simple Rocket Domain

- Verification example: we want to make sure that agents 1 and 3 can move the cargo to the other location.
- $\text{caL} \rightarrow \langle 1, 3 \rangle \diamond \text{caP} \land \text{caP} \rightarrow \langle 1, 3 \rangle \diamond \text{caL}$
Example: Simple Rocket Domain

- Verification example: we want to make sure that agents 1 and 3 can move the cargo to the other location.

- \( \text{cal} \rightarrow \langle 1, 3 \rangle \Diamond \text{caP} \land \text{caP} \rightarrow \langle 1, 3 \rangle \Diamond \text{cal} \land \text{caR} \rightarrow (\langle 1, 3 \rangle \Diamond \text{cal} \land \langle 1, 3 \rangle \Diamond \text{caP}) \)
Example: Simple Rocket Domain
Nice results: model checking CTL and ATL is tractable!
8. Model checking

1. Model Checking Time and Strategies

Nice results: model checking CTL and ATL is tractable!

Theorem (Clarke, Emerson & Sistla 1986)

CTL model checking is $P$-complete, and can be done in time linear in the size of the model and the length of the formula.
Nice results: model checking CTL and ATL is tractable!

Theorem (Clarke, Emerson & Sistla 1986)

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

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!
So... Let’s model-check!

Not as easy as it seems...
Some Complexity Classes

- **P**: problems solvable in **polynomial time** by a deterministic Turing machine
Some Complexity Classes

- **P**: problems solvable in *polynomial time* by a deterministic Turing machine
- **NP**: problems solvable in *polynomial time* by a non-deterministic Turing machine
Some Complexity Classes

- **P**: problems solvable in polynomial time by a deterministic Turing machine
- **NP**: problems solvable in polynomial time by a non-deterministic Turing machine
- **Σ^n_1 / Π^n_1 / Δ^n_1**: problems solvable in polynomial time with use of adaptive queries to an **n-level oracle**
Some Complexity Classes

- **P**: problems solvable in polynomial time by a deterministic Turing machine
- **NP**: problems solvable in polynomial time by a non-deterministic Turing machine
- **Σₙ^P / Πₙ^P / Δₙ^P**: problems solvable in polynomial time with use of adaptive queries to an \( n \)-level oracle
- **PSPACE**: problems solvable by queries to a multilevel oracle with unbounded “height”
Some Complexity Classes

- **P**: problems solvable in polynomial time by a deterministic Turing machine
- **NP**: problems solvable in polynomial time by a non-deterministic Turing machine
- \( \Sigma^P_n / \Pi^P_n / \Delta^P_n \): problems solvable in polynomial time with use of adaptive queries to an \( n \)-level oracle
- **PSPACE**: problems solvable by queries to a multilevel oracle with unbounded “height”
- **EXPTIME**: problems solvable in exponential time
Some Complexity Classes

- **P**: problems solvable in polynomial time by a deterministic Turing machine
- **NP**: problems solvable in polynomial time by a non-deterministic Turing machine
- **Σ^n_P / Π^n_P / Δ^n_P**: problems solvable in polynomial time with use of adaptive queries to an \( n \)-level oracle
- **PSPACE**: problems solvable by queries to a multilevel oracle with unbounded “height”
- **EXPTIME**: problems solvable in exponential time

What is this about?
Some Complexity Classes

- **P**: problems solvable in polynomial time by a deterministic Turing machine
- **NP**: problems solvable in polynomial time by a non-deterministic Turing machine
- **Σ^P_n/Π^P_n/Δ^P_n**: problems solvable in polynomial time with use of adaptive queries to an $n$-level oracle
- **PSPACE**: problems solvable by queries to a multilevel oracle with unbounded “height”
- **EXPTIME**: problems solvable in exponential time

What is this about?

**Scalability!**
### Complexity of Model Checking Temporal and Strategic Logics

<table>
<thead>
<tr>
<th>Logic</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>m, l</td>
<td>m, l</td>
</tr>
<tr>
<td>CTL</td>
<td>P-complete</td>
</tr>
<tr>
<td>LTL</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>CTL*</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>ATL</td>
<td>P-complete</td>
</tr>
<tr>
<td>ATL*</td>
<td>PSPACE-complete</td>
</tr>
</tbody>
</table>
Complexity of Model Checking Temporal and Strategic Logics

<table>
<thead>
<tr>
<th>Logic</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m, l$</td>
<td>$P$-complete</td>
</tr>
<tr>
<td>CTL</td>
<td>$P$-complete</td>
</tr>
<tr>
<td>LTL</td>
<td>$PSPACE$-complete</td>
</tr>
<tr>
<td>CTL*</td>
<td>$PSPACE$-complete</td>
</tr>
<tr>
<td>ATL</td>
<td>$P$-complete</td>
</tr>
<tr>
<td>ATL*</td>
<td>$PSPACE$-complete</td>
</tr>
</tbody>
</table>

For strategies with perfect recall:

<table>
<thead>
<tr>
<th>Logic</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m, l$</td>
<td>$P$-complete</td>
</tr>
<tr>
<td>ATL</td>
<td>$P$-complete</td>
</tr>
<tr>
<td>ATL*</td>
<td>$2EXPTIME$-complete</td>
</tr>
</tbody>
</table>
Nice results: model checking CTL and ATL is tractable.
Nice results: model checking CTL and ATL is tractable.

But: the result is relative to the size of the model and the formula.
- Nice results: model checking CTL and ATL is tractable.
- But: the result is relative to the size of the model and the formula
- Well known catch (CTL): size of models is exponential wrt a higher-level description
3 agents ... 12 states
Do Agents Make Model Checking Explode?
Do Agents Make Model Checking Explode?

<table>
<thead>
<tr>
<th></th>
<th>$m, l$</th>
<th>$n_{\text{local}}, l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTL</td>
<td>P-complete</td>
<td></td>
</tr>
<tr>
<td>LTL</td>
<td>PSPACE-complete</td>
<td></td>
</tr>
<tr>
<td>CTL*</td>
<td>PSPACE-complete</td>
<td></td>
</tr>
<tr>
<td>ATL</td>
<td>P-complete</td>
<td></td>
</tr>
<tr>
<td>ATL*</td>
<td>PSPACE-complete</td>
<td></td>
</tr>
</tbody>
</table>
### Do Agents Make Model Checking Explode?

<table>
<thead>
<tr>
<th></th>
<th>$m, l$</th>
<th>$n_{\text{local}}, l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTL</td>
<td>P-complete</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>LTL</td>
<td>PSPACE-complete</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>CTL*</td>
<td>PSPACE-complete</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>ATL</td>
<td>P-complete</td>
<td></td>
</tr>
<tr>
<td>ATL*</td>
<td>PSPACE-complete</td>
<td></td>
</tr>
</tbody>
</table>
Do Agents Make Model Checking Explode?

<table>
<thead>
<tr>
<th></th>
<th>$m, l$</th>
<th>$n_{local}, l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTL</td>
<td>P-complete</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>LTL</td>
<td>PSPACE-complete</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>CTL*</td>
<td>PSPACE-complete</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>ATL</td>
<td>P-complete</td>
<td>EXPTIME-complete</td>
</tr>
<tr>
<td>ATL*</td>
<td>PSPACE-complete</td>
<td>EXPTIME-complete</td>
</tr>
</tbody>
</table>
Further Problems

- How is the size of a model defined?
Further Problems

- How is the size of a model defined?

  Size of $M = \text{number of transitions in } M$

- What if we define it as the number of states?
Further Problems

- **How is the size of a model defined?**
  
  Size of $M = \text{number of transitions in } M$

- **What if we define it as the number of states?**

- **For CTL:** $m = O(n^2) \not\rightarrow \text{no problem}$
Further Problems

- How is the size of a model defined?
  
  Size of $M = \text{number of transitions in } M$

- What if we define it as the number of states?

- For CTL: $m = O(n^2) \Rightarrow \text{no problem}$

- For ATL: transitions are labeled

- $m$ is not bound by $n^2$!
3 agents . . . 12 states, 216 transitions
Do Agents Make Model Checking Explode?

- Observation: the number of transitions can be exponential in the number of agents
- $m = O(nd^k)$
- $m$: transitions, $n$: states, $d$: actions (decisions), $k$: agents
Do Agents Make Model Checking Explode?

- Observation: the number of transitions can be exponential in the number of agents
  \[ m = O(nd^k) \]
  - \( m \): transitions, \( n \): states, \( d \): actions (decisions), \( k \): agents

- What about model checking?
Do Agents Make Model Checking Explode?

- Observation: the number of transitions can be exponential in the number of agents

- \( m = O(nd^k) \)

- \( m: \text{ transitions}, \ n: \text{ states}, \ d: \text{ actions (decisions)}, \ k: \text{ agents} \)

- What about model checking?

Theorem (Jamroga & Dix 2005; Laroussinie, Markey & Oreiby 2006)

ATL model checking is \( \Delta^P_2 \)-complete with respect to the number of states and agents.
## Summary of Complexity Results

<table>
<thead>
<tr>
<th>Logic</th>
<th>$m, l$</th>
<th>$n, k, l$</th>
<th>$n_{local}, l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTL*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATL*</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Summary of Complexity Results

<table>
<thead>
<tr>
<th></th>
<th>$m, l$</th>
<th>$n, k, l$</th>
<th>$n_{local}, l$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CTL</strong></td>
<td><strong>P</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LTL</strong></td>
<td><strong>PSPACE</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>CTL</strong>*</td>
<td><strong>PSPACE</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ATL</strong></td>
<td><strong>P</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ATL</strong>*</td>
<td><strong>PSPACE</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Summary of Complexity Results

<table>
<thead>
<tr>
<th></th>
<th>$m, l$</th>
<th>$n, k, l$</th>
<th>$n_{local}, l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTL</td>
<td>$P$</td>
<td></td>
<td>PSPACE</td>
</tr>
<tr>
<td>LTL</td>
<td>PSPACE</td>
<td></td>
<td>PSPACE</td>
</tr>
<tr>
<td>CTL*</td>
<td>PSPACE</td>
<td></td>
<td>PSPACE</td>
</tr>
<tr>
<td>ATL</td>
<td>$P$</td>
<td></td>
<td>EXPTIME</td>
</tr>
<tr>
<td>ATL*</td>
<td>PSPACE</td>
<td></td>
<td>EXPTIME</td>
</tr>
</tbody>
</table>
# Summary of Complexity Results

<table>
<thead>
<tr>
<th></th>
<th>$m, l$</th>
<th>$n, k, l$</th>
<th>$n_{local}, l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTL</td>
<td>$\mathbb{P}$</td>
<td>$\mathbb{P}$</td>
<td>PSPACE</td>
</tr>
<tr>
<td>LTL</td>
<td>PSPACE</td>
<td>PSPACE</td>
<td>PSPACE</td>
</tr>
<tr>
<td>CTL*</td>
<td>PSPACE</td>
<td>PSPACE</td>
<td>PSPACE</td>
</tr>
<tr>
<td>ATL</td>
<td>$\mathbb{P}$</td>
<td>$\Delta^P_2$</td>
<td>EXPTIME</td>
</tr>
<tr>
<td>ATL*</td>
<td>PSPACE</td>
<td>EXPTIME</td>
<td>EXPTIME</td>
</tr>
</tbody>
</table>
Looking for Moral

Main message:

- Complexity is very sensitive to the context!
Main message:

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

Still, people do automatic model checking!
Even if model checking appears very easy, it can be very hard.

Still, people do automatic model checking!

- LTL: SPIN
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 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.
8.2 Imperfect Information
Model Checking Imperfect Information Games

Recall: $\langle A \rangle_{ir}$ are not fixpoint operators any more

Conjecture

Strategy for $A$ cannot be synthesized incrementally.
Model Checking Imperfect Information Games

Recall: $\langle A \rangle_{ir}$ are not fixpoint operators any more

Conjecture

Strategy for $A$ cannot be synthesized incrementally.

Indeed...
Model Checking Imperfect Information Games

Recall: $\langle A \rangle_{ir}$ are not fixpoint operators any more

**Conjecture**
Strategy for $A$ cannot be synthesized incrementally.

Indeed...

**Theorem (Schobbens 2004; Jamroga & Dix 2006)**
Model checking $\text{ATL}_{ir}$ is $\Delta_2$-complete in the number of transitions in the model and the length of the formula.
8. Model checking

2. Imperfect Information

Proof Idea: Inclusion in $\Delta_2$

Let $mctl(\varphi, M)$ be a CTL model checker that returns the set of all states that satisfy $\varphi$ in $M$

$mcheck(M, q, \langle A \rangle \Box \psi)$:

1. Run $mcheck(\psi, M, q)$ for every $q \in St$, and label the states in which the answer was “yes” with an additional proposition yes (not used elsewhere).

2. Guess the best strategy of $A$, and “trim” model $M$ by removing all the transitions inconsistent with the strategy (yielding a sparser model $M'$).

3. Return “yes” if $\text{img}(q, \sim^E_A) \subseteq mctl(A \Box \text{yes}, M')$, and “no” otherwise.

Other cases: analogous
Proof Idea: Hardness (by reduction of SNSAT)

**Definition (SNSAT)**

**Input:**
- $z_1 \equiv \exists X_1 \phi_1(z_1, X_1)$
- $z_2 \equiv \exists X_2 \phi_2(z_1, z_2, X_2)$
- ..... 
- $z_p \equiv \exists X_p \phi_p(z_1, \ldots, z_{p-1}, X_p)$.

**Output:** The truth value of $z_p$. 
8. Model checking

2. Imperfect Information

Proof Idea: Hardness (by reduction of SNSAT)

Definition (SNSAT)

Input:
- \( z_1 \equiv \exists X_1 \varphi_1(z_1, X_1) \)
- \( z_2 \equiv \exists X_2 \varphi_2(z_1, z_2, X_2) \)
- ......
- \( z_p \equiv \exists X_p \varphi_p(z_1, \ldots, z_{p-1}, X_p) \).

Output: The truth value of \( z_p \).

Lemma 8.1

Let \( \Phi_1 \equiv \langle \langle v \rangle \rangle_{ir} (\neg \text{neg}) \cup \text{yes} \),
\( \Phi_i \equiv \langle \langle v \rangle \rangle_{ir} (\neg \text{neg}) \cup (\text{yes} \lor (\text{neg} \land A \bigcirc \neg \Phi_{i-1})) \).

Now, for all \( r \): \( z_r \) is true iff \( M_r, q_0^r \models \Phi_r \).
Proof Idea: Hardness
Model Checking Imperfect Information Games

Corollary

Imperfect information strategies cannot be synthesized incrementally: we cannot do better than guess the whole strategy and check if it succeeds.
Model Checking Imperfect Information Games

**Corollary**

Imperfect information strategies cannot be synthesized incrementally: *we cannot do better than guess the whole strategy and check if it succeeds.*

Imperfect information makes model checking harder!
8. Model checking

2. Imperfect Information

Model Checking Imperfect Information Games

Corollary

Imperfect information strategies cannot be synthesized incrementally: we cannot do better than guess the whole strategy and check if it succeeds.

Imperfect information makes model checking harder!

Or...?
## Summary of Model Checking Results

<table>
<thead>
<tr>
<th></th>
<th>$m, l$</th>
<th>$n, k, l$</th>
<th>$n_{\text{local}}, k, l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\text{ATL}_{\text{ir}}/\text{CSL}$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Summary of Model Checking Results

<table>
<thead>
<tr>
<th></th>
<th>$m, l$</th>
<th>$n, k, l$</th>
<th>$n_{\text{local}}, k, l$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CTL</strong></td>
<td>$P$</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ATL</strong></td>
<td>$P$</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>ATL_{ir}/CSL</strong></td>
<td>$\Delta^P_2$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Summary of Model Checking Results

<table>
<thead>
<tr>
<th></th>
<th>$m, l$</th>
<th>$n, k, l$</th>
<th>$n_{\text{local}}, k, l$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CTL</strong></td>
<td>$P$</td>
<td>$P$</td>
<td></td>
</tr>
<tr>
<td><strong>ATL</strong></td>
<td>$P$</td>
<td>$\Delta_3^P$</td>
<td></td>
</tr>
<tr>
<td><strong>ATL_{ir/CSL}</strong></td>
<td>$\Delta_2^P$</td>
<td>$\Delta_3^P$</td>
<td></td>
</tr>
</tbody>
</table>
## Summary of Model Checking Results

<table>
<thead>
<tr>
<th>Logics</th>
<th>$m, l$</th>
<th>$n, k, l$</th>
<th>$n_{local}, k, l$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTL</td>
<td>P</td>
<td>P</td>
<td>PSPACE</td>
</tr>
<tr>
<td>ATL</td>
<td>P</td>
<td>$\Delta_3^P$</td>
<td>EXPTIME</td>
</tr>
<tr>
<td>ATL$_{ir}$/CSL</td>
<td>$\Delta_2^P$</td>
<td>$\Delta_3^P$</td>
<td>PSPACE</td>
</tr>
</tbody>
</table>
# Summary of Model Checking Results

<table>
<thead>
<tr>
<th></th>
<th>$m, l$</th>
<th>$n, k, l$</th>
<th>$n_{local}, k, l$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ATL</strong></td>
<td>$P$ [3]</td>
<td>$\Delta^P_3$ [5,8]</td>
<td>EXPTIME [6,7]</td>
</tr>
</tbody>
</table>

The Message Again...

- Complexity is **very** sensitive to the context!
- In particular, **the way we define the input, and measure its size**, is crucial.
8.3 The Phantom Result
Between Perception and Recall

<table>
<thead>
<tr>
<th>logic</th>
<th>$ir$</th>
<th>$iR$</th>
<th>$Ir$</th>
<th>$IR$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\langle \Gamma \rangle - ATL$</td>
<td>$NP$</td>
<td>$U\ [11]$</td>
<td>$n \cdot l\ [2]$</td>
<td>$n \cdot l\ [2]$</td>
</tr>
<tr>
<td>$ATL^+$</td>
<td>$\Delta_3P$</td>
<td>$U\ [11]$</td>
<td>$\Delta_3P$</td>
<td>$\Delta_3P$</td>
</tr>
</tbody>
</table>

$NP$ complete for nondeterministic polynomial time

$\Delta_2P = P^{NP}$ complete for polynomial calls to an $NP$ oracle

$\Delta_3P = P^{NP^{NP}}$ complete for polynomial calls to a $\Sigma_2P$ oracle

$EXP$ complete for deterministic exponential time

$DEXP$ complete for deterministic doubly exponential time

$U$ undecidable

$l$ size of the formula

$n$ size of the model
The Undecidability “Result”

- Most cite it from (Alur et al., 1997–2002)
- Alur et al. cite Yannakakis ("Synchronous multi-player games with incomplete information are undecidable", 1997)
- **Personal communication!**
- No proof has been published (nor has the result been formally stated)
Relevant Existing Results

(Peterson & Reif, 1979):
Relevant Existing Results

- (Peterson & Reif, 1979):
  - Solving games with imperfect information and perfect recall is decidable in the case of a single proponent.
  - Solving games with imperfect information and perfect recall is undecidable in the case of a team of proponents.
Relevant Existing Results

- (Peterson & Reif, 1979):
  - Solving games with imperfect information and perfect recall is decidable in the case of a single proponent.
  - Solving games with imperfect information and perfect recall is undecidable in the case of a team of proponents.

- But: their games are defined via Turing machines, while in “our” games are close to finite automata.
Relevant Existing Results

- (Pnueli & Rosner, 1990):
Relevant Existing Results

- (Pnueli & Rosner, 1990):
  - **Realizability problem** for distributed systems is undecidable

- The setting very close to ours. Difference: “winning conditions” are defined via LTL specifications, so we have winning **paths** rather than states. In particular, the reduction of the halting problem for deterministic Turing Machines to the realizability problem (that proves undecidability of the problem) employs LTL formulae that are not expressible in CTL
Relevant Existing Results

- (Van der Meyden and Shilov, 1999):
Relevant Existing Results

- (Van der Meyden and Shilov, 1999):
  - Model checking $\text{LTL}+\text{K}$ with perfect recall is decidable (with a nonelementary lower bound)
  - Model checking $\text{LTL}+\text{K}+\text{C}$ with perfect recall is undecidable
Relevant Existing Results

- (Van der Meyden and Shilov, 1999):
  - Model checking $\text{LTL+K}$ with perfect recall is decidable (with a non-elementary lower bound)
  - Model checking $\text{LTL+K+C}$ with perfect recall is undecidable

- (Garanina, Kalinina and Shilov, 2004):
Relevant Existing Results

- (Van der Meyden and Shilov, 1999):
  - Model checking $\text{LTL+K}$ with perfect recall is decidable (with a nonelementary lower bound)
  - Model checking $\text{LTL+K+C}$ with perfect recall is undecidable

- (Garanina, Kalinina and Shilov, 2004):
  - Model checking $\text{CTL+K}$ with perfect recall is decidable (with a nonelementary lower bound)
  - Model checking $\text{CTL+K+C}$ with perfect recall is undecidable
8.4 References

Axiom. of Coal. Games
9.1 Coalitional Games
Coalitional Games

The difference between non-cooperative games and coalitional games is that the former takes possible actions of individual players as primary, while the latter takes possible actions of coalitions as primary.
Non-coop. Game: Individual Actions Primary

<table>
<thead>
<tr>
<th>Ann</th>
<th>Bill</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cooperate</td>
</tr>
<tr>
<td>Cooperate</td>
<td>Ann:-1, Bill:-1</td>
</tr>
<tr>
<td>Defect</td>
<td>Ann:0, Bill:-4</td>
</tr>
</tbody>
</table>

Bill

We can derive possible actions of coalitions, and thus coalitional power, from the individual actions.
Non-coop. Game: Individual Actions Primary

\[
\begin{array}{c|cc}
\text{Bill} & \text{Cooperate} & \text{Defect} \\
\hline
\text{Ann} & \text{Cooperate} & \text{Defect} \\
\text{Cooperate} & \text{Ann:-1, Bill:-1} & \text{Ann:-4, Bill: 0} \\
\text{Defect} & \text{Ann:0, Bill:-4} & \text{Ann:-3, Bill: -3} \\
\end{array}
\]

\textbf{ATL/CL:}

\[
\lll\{Ann\}\rrl B \geq 3
\]

We can derive possible actions of coalitions, and thus coalitional power, from the individual actions:

\[
\lll\{Ann, Bill\}\rrl (A = 1 \land B = 1)
\]
Coalitional Game: Coalitional Actions Primary

Example: Three-player majority game:

- Three persons
- One cake
- Any majority group (two or three) controls the division of the cake to the members of the group
- Each person cares (only) about how much cake he gets

![Diagram of a three-person game with two outcomes]
Definition 9.1 (Coalitional Game (with Transferable Payoff))

A coalitional game (with transferable payoff) is a tuple $(N, \Omega, V, \{\sqsupseteq_i\}_{i \in N})$:

- $N$ is the set of players
- $\Omega$ is the set of outcomes
- $V$ assigns a set of choices $V(C) \subseteq \Omega$ to each non-empty coalition $C \subseteq N$
- For each $i$, $\sqsupseteq_i$ is a preference relation over the outcomes
  - Usually assumed to be reflexive, transitive and complete
  - We write $\sqsubseteq_i$ for the strict variant
  - Is often described by a utility function $u_i$ for each player $i$ over the outcomes: $\omega \sqsupseteq_i \omega'$ iff $u_i(\omega) \geq u_i(\omega')$
Coalitional Game: Example

The cake game:

- \( N = \{Ann, Bill, Cath\} \)
- \( \Omega \): the collection of possible ways to divide a cake between Ann and Bill, between Ann and Cath, between Bill and Cath, and between Ann and Bill and Cath
  - \( \Omega = \{(A = 10\%, B = 90\%), (B = 50\%, C = 50\%), (A = 20\%, B = 30\%, C = 50\%), \ldots\} \)
- \( V(C) = \)
  \[
  \begin{cases}
  \text{all divisions of the cake among } C & |C| \geq 2 \\
  \emptyset & \text{otherwise}
  \end{cases}
  \]
- \( \omega_1 \equiv_{Ann} \omega_2 \) iff Ann gets at least as much cake in \( \omega_1 \) as in \( \omega_2 \), etc.
Coalitional Games with Transferable Payoff

Definition 9.2

A coalitional game with transferable payoff is a pair \((N, v)\):

- \(N\) is the set of players
- \(v\) assigns a real number \(v(C)\) to each non-empty coalition \(C \subseteq N\); the worth of \(C\)
Coalitional Games with Transferable Payoff

Definition 9.2

Coalitional Game with Transferable Payoff

A coalitional game with transferable payoff is a pair \((N, v)\):

- \(N\) is the set of players
- \(v\) assigns a real number \(v(C)\) to each non-empty coalition \(C \subseteq N\); the worth of \(C\)

Games WTP can be seen as a special class of games WOTP
Coalitional Games with Transferable Payoff

Definition 9.2

A coalitional game with transferable payoff is a pair \((N, v)\):

- \(N\) is the set of players
- \(v\) assigns a real number \(v(C)\) to each non-empty coalition \(C \subseteq N\); the worth of \(C\)

Games WTP can be seen as a special class of games WOTP
- But games WOTP are more general
Coalitional Games with Transferable Payoff

Definition 9.2

A coalitional game with transferable payoff is a pair \((N, v)\):
- \(N\) is the set of players
- \(v\) assigns a real number \(v(C)\) to each non-empty coalition \(C \subseteq N\); the worth of \(C\)

- Games WTP can be seen as a special class of games WOTP
- But games WOTP are more general
- We will use games WOTP
Coalitional Games with Transferable Payoff

Definition 9.2

A coalitional game with transferable payoff is a pair \((N, v)\):

- \(N\) is the set of players
- \(v\) assigns a real number \(v(C)\) to each non-empty coalition \(C \subseteq N\); the worth of \(C\)

- Games WTP can be seen as a special class of games WOTP
- But games WOTP are more general
- We will use games WOTP
- Henceforth: by “coalitional game” we mean “coalitional game WOTP”.
A solution concept assigns a set of outcomes to each game.

General idea: like in non-cooperative games: what are the stable outcomes?

Stability: no coalition can profit from deviating.
Solution Concepts for Coalitional Games

- A solution concept assigns a set of outcomes to each game.
- General idea: like in non-cooperative games: what are the stable outcomes?
- Stability: no coalition can profit from deviating.
- Some important concepts:
  - The core
  - Stable sets
  - The bargaining set
The Core

Definition 9.3 (The Core)

The core of a coalitional game is the set of outcomes \( \omega \in V(N) \) for which there is no coalition \( C \) with an outcome \( \omega' \in V(C) \) such that \( \omega' \succ_i \omega \) for all \( i \in C \).
9. Axiom. of Coal. Games

1. Coalitional Games

The Core

Definition 9.3 (The Core)

The core of a coalitional game is the set of outcomes \( \omega \in V(N) \) for which there is no coalition \( C \) with an outcome \( \omega' \in V(C) \) such that \( \omega' \succ_i \omega \) for all \( i \in C \).

What is the core of the cake game?
The Core

Definition 9.3 (The Core)

The core of a coalitional game is the set of outcomes \( \omega \in V(N) \) for which there is no coalition \( C \) with an outcome \( \omega' \in V(C) \) such that \( \omega' \succ_i \omega \) for all \( i \in C \).

What is the core of the cake game?

Key property of a coalitional game: is the core empty?
Stable Sets

- Idea: an outcome is stable if no (sub)coalition has an incentive to deviate and form a stable coalition (recursive!)
- From von Neumann and Morgenstern, 1944
- A stable set is a set of outcomes
- A game may have more than one stable set
- .. but must not have any
- Characterised by imputations and objections
9. Axiom. of Coal. Games

1. Coalitional Games

Imputation

Definition 9.4 (Imputation)
An imputation is an outcome $\omega \in V(N)$ that for each agent $i$ is as least as good as any outcome the singleton coalition $\{i\}$ can choose on his own.
Objection

Definition 9.5 (Objection)

An imputation $\omega$ is a $C$-objection to an imputation $\omega'$ if every agent in $C$ prefers $\omega$ over $\omega'$ and the coalition $C$ can choose an outcome which for every agent in $C$ is as least as good as $\omega$. $\omega$ is an objection to $\omega'$ if $\omega$ is a $C$-objection to $\omega'$ for some coalition $C$. 
Stable Set

Definition 9.6 (Stable Set)

A set of imputations $Y$ is a stable set if it satisfies:

**Internal stability** If $\omega \in Y$, there is no objection to $\omega$ in $Y$.

**External stability** If $\omega \not\in Y$, there is an objection to $\omega$ in $Y$. 
The Bargaining Set

- A set of imputations
- Unique
- Always exists
- Defined in terms of objections and counterobjections – but the concept of objection is different from the stable sets case
- Will introduce it formally later
9.2 Coalitional Game Logic
Goal

- We want to be able to reason about coalitional games in a formal logic
- In particular: characterise solution concepts
Coalitional Game Logic

- We have already used the modality $\langle C \rangle$ to reason about coalitional ability in non-cooperative games.
- It is natural and straightforward to interpret this modality by the $V$ function in coalitional games.
- Additional assumptions on propositions in the language:
  - $\omega$, where $\omega$ is (the name of) an outcome in $\Omega$: meaning that the current outcome is $\omega$.
  - $\omega \geq_i \omega'$: meaning that agent $i$ weakly prefers outcome $\omega$ over $\omega'$. 
Coalitional Game Logic

We have already used the modality $\langle \langle C \rangle \rangle$ to reason about coalitional ability in non-cooperative games.

It is natural and straightforward to interpret this modality by the $V$ function in coalitional games.

Additional assumptions on propositions in the language:
- $\omega$, where $\omega$ is (the name of) an outcome in $\Omega$: meaning that the current outcome is $\omega$.
- $\omega \succeq_i \omega'$: meaning that agent $i$ weakly prefers outcome $\omega$ over $\omega'$.

Let $\Gamma$ be a coalitional game.

\[
\Gamma \models \omega \succeq_i \omega' \iff \omega \sqsupseteq_i \omega' \\
\Gamma \models \langle \langle C \rangle \phi \iff \exists \omega \in V(C), \omega \models \phi \\
\omega \models \omega' \iff \omega = \omega'
\]
The Core

The core of a coalitional game is the set of outcomes $\omega \in V(N)$ for which there is no coalition $C$ with an outcome $\omega' \in V(C)$ such that $\omega' \succ_i \omega$ for all $i \in C$.

$\omega$ is a member of the core (assuming finite $\Omega$):

$$CM(\omega) \equiv \langle N \rangle \omega \land \neg \left[ \bigvee_{C \subseteq N} \bigvee_{\omega' \in \Omega} (\langle C \rangle \omega') \land \bigwedge_{i \in C} (\omega' \succ_i \omega) \right]$$
The Core

The core of a coalitional game is the set of outcomes \( \omega \in V(N) \) for which there is no coalition \( C \) with an outcome \( \omega' \in V(C) \) such that \( \omega' \succ_i \omega \) for all \( i \in C \).

\( \omega \) is a member of the core (assuming finite \( \Omega \)):

\[
CM(\omega) \equiv \langle N \rangle \omega \land \neg \left( \bigvee_{C \subseteq N} \bigvee_{\omega' \in \Omega} (\langle C \rangle \omega') \land \bigwedge_{i \in C} (\omega' \succ_i \omega) \right)
\]

The core is non-empty:

\[
CNE \equiv \bigvee_{\omega \in \Omega} CM(\omega)
\]
Imputation

An imputation is an outcome $\omega \in V(N)$ that for each agent $i$ is as least as good as any outcome the singleton coalition $\{i\}$ can choose on his own.

$$IMP(\omega) \equiv \langle N \rangle \omega \land \bigwedge_{\omega' \in \Omega} \bigwedge_{i \in N} (\langle \{i\} \rangle \omega' \rightarrow \omega \succeq_i \omega')$$
Objection

An imputation $\omega$ is a $C$-objection to an imputation $\omega'$ if every agent in $C$ prefers $\omega$ over $\omega'$ and the coalition $C$ can choose an outcome which for every agent in $C$ is as least as good as $\omega$. $\omega$ is an objection to $\omega'$ if $\omega$ is a $C$-objection to $\omega'$ for some coalition $C$.

$$OBJ(\omega, \omega', C) \equiv (\bigwedge_{i \in C} \omega \succeq_i \omega') \land \bigvee_{\omega'' \in \Omega} (\langle C \rangle \omega'' \land \bigwedge_{i \in C} \omega'' \succeq_i \omega)$$
Stable Set

A set of imputations $Y$ is a stable set if it satisfies:

**Internal stability** If $\omega \in Y$, there is no objection to $\omega$ in $Y$.

**External stability** If $\omega \notin Y$, there is an objection to $\omega$ in $Y$.

\[
STABLE(Y) \equiv \\
\Lambda_{\omega \in Y} IMP(\omega) \\
\wedge \left( \Lambda_{\omega \in Y} \Lambda_{C \subseteq N} \Lambda_{\omega' \in Y} \neg OBJ(\omega', \omega, C) \right) \\
\wedge \left( \Lambda_{\omega \in \Omega \setminus Y} IMP(\omega) \rightarrow \left( \bigvee_{C \subseteq N} \bigvee_{\omega' \in Y} OBJ(\omega', \omega, C) \right) \right)
\]
The Bargaining Set

$\omega'$ is an objection of $C$ to $\omega$:

$$OBJB(\omega', C, \omega) \equiv \langle C \rangle \omega' \land \bigwedge_{k \in C} \omega' \succ_k \omega$$

There exists a counterobjection to the objection $\omega'$ of $C$ to $\omega$, where $i \in C$ and $j \notin C$:

$$COUNTER(\omega', C, i, j, \omega) \equiv \bigvee_{v \in \Omega} \bigvee_{D' \subseteq N \setminus \{i\}} (\langle D' \cup \{j\} \rangle v \land (\bigwedge_{k \in (D' \cup \{j\}) \setminus C} v \succeq_k \omega) \land (\bigwedge_{k \in (D' \cup \{j\}) \cap C} v \succeq_k \omega'))$$
The Bargaining Set

Outcome $\omega$ is in the bargaining set:

$$INBARG(\omega) \equiv IMP(\omega) \land \bigwedge_{C \subseteq N} \bigwedge_{i \in C} \bigwedge_{j \in N \setminus C} \bigwedge_{\omega' \in \Omega} \left[ OBJB(\omega', C, \omega) \rightarrow COUNTER(\omega', C, i, j, \omega) \right]$$

$$BS(Y) = \bigwedge_{\omega \in Y} INBARG(\omega) \land \bigwedge_{\omega \in \Omega \setminus Y} \neg INBARG(\omega)$$
This Coalitional Game Logic:
- is very expressive (for finite games)
- can characterise solution concepts
This Coalitional Game Logic:

- is very expressive (for finite games)
- can characterise solution concepts

However, the characterisations:

- do not work for infinite games (games with infinitely many outcomes)
- are not very succinct
- depend on $\Omega$ and are thus different for games with different sets of outcomes
Recall the def. of a coalitional game:

\[(N, \Omega, V, \{\equiv_i\}_{i \in N})\]
Recall the definition of a coalitional game:

\[(N, \Omega, V, \{\equiv_i\}_{i \in N})\]

In Coalitional Game Logic we used \(V\) to interpret \(\langle C \rangle\), and atomic propositions for \(\equiv_i\).
Recall the def. of a coalitional game:

\[ (N, \Omega, V, \{\equiv_i\}_{i \in N}) \]

In Coalitional Game Logic we used \(V\) to interpret \(\langle C\rangle\), and atomic propositions for \(\equiv_i\)

**Observation:** there is “more structure” in \(\equiv_i\)!
Recall the def. of a coalitional game:

\[(N, \Omega, V, \{\equiv_i\}_{i \in N})\]

In Coalitional Game Logic we used \(V\) to interpret \(\langle C \rangle\), and atomic propositions for \(\equiv_i\)

**Observation:** there is “more structure” in \(\equiv_i\)!

**Modal Coalitional Game Logic:** we will use \(\equiv_i\) to interpret \(\langle C \rangle\), and atomic propositions for \(V\).
Modal Coalitional Game Logic (MCGL)

Main constructs ($C \subseteq N$):

\[ \langle C \rangle \varphi \]

meaning: (all agents in) $C$ prefers $\varphi$
Modal Coalitional Game Logic (MCGL)

Main constructs \((C \subseteq N)\):

\[
\langle C \rangle \varphi
\]

meaning: (all agents in) \(C\) prefers \(\varphi\)

and

\[
p_C
\]

meaning: \(C\) can choose the current outcome
Modal Coalitional Game Logic (MCGL)

Main constructs \((C \subseteq N)\):

\[ \langle C \rangle \varphi \]

meaning: (all agents in) \(C\) prefers \(\varphi\)

and

\[ p_C \]

meaning: \(C\) can choose the current outcome

Henceforth: use

\[ C = 2^N \setminus \emptyset \]

to denote the set of coalitions
Formal Language

Let

\[ \Theta = \Theta' \cup \{ p_C : C \subseteq N \} \]

where \( \Theta' \) is a countably infinite set of atomic propositions.
Formal Language

Let

$$\Theta = \Theta' \cup \{ p_C : C \subseteq N \}$$

where \( \Theta' \) is a countably infinite set of atomic propositions

The MCGL language (will add more later):

$$\phi ::= p \mid \langle C \rangle \phi \mid \langle C^s \rangle \phi \mid \neg \phi \mid \phi_1 \land \phi_2 \mid \cdots$$

where \( p \in \Theta, C \subseteq N, i \in N \). Derived: \([\cdot],[\cdot^s]\) are the duals of \(\langle \cdot \rangle, \langle \cdot^s \rangle\), respectively.
Interpretation

Let $\Gamma = (N, \Omega, V, \Box_1, \ldots, \Box_m)$ be a coalitional game, let $\pi$ be a valuation of $\Theta'$ in $\Omega$, and let $w \in \Omega$.

- $\Gamma, \pi, w \models p_C$ iff $w \in V(C)$
- $\Gamma, \pi, w \models p$ iff $w \in \pi(p)$, when $p \in \Theta'$
- $\Gamma, \pi, w \models \langle C \rangle \phi$ iff there is a $v$ such that for every $i \in C$, $v \Box_i w$, and $\Gamma, \pi, v \models \phi$
Interpretation

Let $\Gamma = (N, \Omega, V, \sqsupseteq_1, \ldots, \sqsupseteq_m)$ be a coalitional game, let $\pi$ be a valuation of $\Theta'$ in $\Omega$, and let $w \in \Omega$.

- $\Gamma, \pi, w \models p_C$ iff $w \in V(C)$
- $\Gamma, \pi, w \models p$ iff $w \in \pi(p)$, when $p \in \Theta'$
- $\Gamma, \pi, w \models \langle C \rangle \phi$ iff there is a $v$ such that for every $i \in C$, $v \sqsupseteq_i w$, and $\Gamma, \pi, v \models \phi$
- $\Gamma, \pi, w \models \langle C^s \rangle \phi$ iff there is a $v$ such that for every $i \in C$, $v \sqsupseteq_i w$ and not $w \sqsupseteq_i v$, and $\Gamma, \pi, v \models \phi$
Interpretation

Let $\Gamma = (N, \Omega, V, \sqsupseteq_1, \ldots, \sqsupseteq_m)$ be a coalitional game, let $\pi$ be a valuation of $\Theta'$ in $\Omega$, and let $w \in \Omega$.

- $\Gamma, \pi, w \models p_C$ iff $w \in V(C)$
- $\Gamma, \pi, w \models p$ iff $w \in \pi(p)$, when $p \in \Theta'$
- $\Gamma, \pi, w \models \langle C \rangle \phi$ iff there is a $v$ such that for every $i \in C$, $v \sqsupseteq_i w$, and $\Gamma, \pi, v \models \phi$
- $\Gamma, \pi, w \models \langle C^s \rangle \phi$ iff there is a $v$ such that for every $i \in C$, $v \sqsupseteq_i w$ and not $w \sqsupseteq_i v$, and $\Gamma, \pi, v \models \phi$

Let us write:

$$\Gamma, w \models \phi \iff \Gamma, \pi, w \models \phi \text{ for all } \pi$$

$$\Gamma \models \phi \iff \Gamma, w \models \phi \text{ for all } w$$
Characterising the Core

The core of a coalitional game is the set of outcomes $\omega \in V(N)$ for which there is no coalition $C$ with an outcome $\omega' \in V(C)$ such that $\omega' \succ_i \omega$ for all $i \in C$. 
Characterising the Core

The core of a coalitional game is the set of outcomes $\omega \in V(N)$ for which there is no coalition $C$ with an outcome $\omega' \in V(C)$ such that $\omega' \succ_i \omega$ for all $i \in C$.

$$MCM \equiv p_N \land \bigwedge_{C \subseteq N} [C^s] \neg p_C$$

**Theorem 9.7**

$\Gamma, \omega \models MCM$ iff $\omega$ is in the core of $\Gamma$
Characterising Imputations

An imputation is an outcome $\omega \in V(N)$ that for each agent $i$ is as least as good as any outcome the singleton coalition $\{i\}$ can choose on his own.
Characterising Imputations

An imputation is an outcome $\omega \in V(N)$ that for each agent $i$ is as least as good as any outcome the singleton coalition $\{i\}$ can choose on his own.

$$MIMP \equiv p_N \land \bigwedge_{i \in N} [C^s] \neg p_i$$

**Theorem 9.8**

$\Gamma, \omega \models MIMP$ iff $\omega$ is an imputation in $\Gamma$
Stable Sets

- Difficult to characterise stable sets and the bargaining set in MCGL
- How to refer to sets of outcomes? Formulae are interpreted in single outcomes, and we can’t refer directly to outcomes in the formula (unlike in CGL).
- Here is a way: the extension

\[ \phi^\Gamma = \{ \omega : \Gamma, \omega \models \phi \} \]

is a set.
- Example: \( MCM\Gamma \) is the core of \( \Gamma \)
Stable Sets

Let

\[ MOBJ(C, \alpha) \equiv MIMP \land \langle C^s \rangle (MIMP \land \alpha \land \langle C \rangle p_C) \]

meaning: \( \Gamma, \omega \models MOBJ(C, \alpha) \) iff \( \omega \) is an imputation and there exists a \( C \)-objection \( \omega' \) to \( \omega \) such that \( \Gamma, \omega' \models \alpha \).
Stable Sets

Let

$$MOBJ(C, \alpha) \equiv MIMP \land \langle C^s \rangle (MIMP \land \alpha \land \langle C \rangle p_C)$$

meaning: $\Gamma, \omega \models MOBJ(C, \alpha)$ iff $\omega$ is an imputation and there exists a $C$-objection $\omega'$ to $\omega$ such that $\Gamma, \omega' \models \alpha$

**Theorem 9.9**

Let $\gamma$ be a formula.

$$\Gamma \models (\gamma \rightarrow MIMP) \land (\gamma \rightarrow \neg \bigvee_{C \subseteq N} MOBJ(C, \gamma)) \land (\neg \gamma \rightarrow \bigvee_{C \subseteq N} MOBJ(C, \gamma))$$

iff $\gamma^\Gamma$ is a stable set in $\Gamma$. 
MCGL: advantages and disadvantages

- Note that we can characterise, e.g., the core also for **infinite** games
- The characterisation is the same for all games over the same set of agents
- But: not as expressive as CGL (for finite games)
9.4 Axiomatisation
Let us try to view this as a normal modal logic – and be very explicit. Henceforth assume a fixed set of agents $N$.

$$\phi ::= p \mid \langle C \rangle \phi \mid \langle C^s \rangle \phi \mid \neg \phi \mid \phi_1 \land \phi_2 \mid \cdots$$

A model would be a tuple:

$$M = (W, \{ R_C : C \in C \}, \{ R^s_C : C \in C \}, \pi)$$

where $\pi$ is a valuation of $\Theta = \Theta' \cup \{ p_C : C \in C \}$
Let us try to view this as a normal modal logic – and be very explicit. Henceforth assume a fixed set of agents $N$.

$$\phi ::= p \mid \langle C' \rangle \phi \mid \langle C^s \rangle \phi \mid \neg \phi \mid \phi_1 \land \phi_2 \mid \cdots$$

A model would be a tuple:

$$M = (W, \{R_C : C \in C\}, \{R^s_C : C \in C\}, \pi)$$

where $\pi$ is a valuation of $\Theta = \Theta' \cup \{p_C : C \in C\}$

And to get correspondence with coalitional games:

- **REFL** $\forall i \in N R_i$ is reflexive
- **TRANS** $\forall i \in N R_i$ is transitive
- **COMPL** $\forall i \in N R_i$ is complete
- **STRICT** $\forall i \in N R^s_i wu$ iff both $R_i wu$ and not $R_i uw$
- **INTERSECTION** $\forall C \in C R_C = \bigcap_{i \in C} R_i$
- **INTERSECTION-STRICT** $\forall C \in C R^s_C = \bigcap_{i \in C} R^s_i$

where we write $R_i$ for $R\{i\}$.
Axioms

**REFL, TRANS:**

- **T** \( [i]p \rightarrow p \)
- **4** \( [i]p \rightarrow [i][i]p \)
Axioms

**REFL, TRANS:**

\[
\begin{array}{|c|c|}
\hline
T & [i]p \rightarrow p \\
4 & [i]p \rightarrow [i][i]p \\
\hline
\end{array}
\]

.. but several of the other properties do not have canonical formulae
Common approach when we have a property $P$, such as \textbf{INTERSECTION}, which are neither modally definable nor has a canonical formula:

- Construct the canonical model
Common approach when we have a property $P$, such as \textbf{INTERSECTION}, which are neither modally definable nor has a canonical formula:

- Construct the canonical model
- \textbf{Transform} it into a model which satisfies the same formulae, but which has the property $P$
Common approach when we have a property $P$, such as INTERSECTION, which are neither modally definable nor has a canonical formula:

- **Construct the canonical model**
- **Transform** it into a model which satisfies the same formulae, but which has the property $P$
- **However:** transformation may be difficult when there are several properties that must be achieved/maintained at the same time – as in our case
Common approach when we have a property \( P \), such as \textsc{intersection}, which are neither modally definable nor has a canonical formula:

- Construct the canonical model
- \textbf{Transform} it into a model which satisfies the same formulae, but which has the property \( P \)
- However: transformation may be difficult when there are several properties that must be achieved/maintained at the same time – as in our case
- Many alternative approaches have been used and studied in detail
Common approach when we have a property $P$, such as \textsc{Intersection}, which are neither modally definable nor has a canonical formula:

- Construct the canonical model
- \textbf{Transform} it into a model which satisfies the same formulae, but which has the property $P$
- However: transformation may be difficult when there are \textit{several properties} that must be achieved/maintained at the same time – as in our case
- Many alternative approaches have been used and studied in detail
- Here: we will use standard techniques combining the \textit{difference modality} with a \textit{step-by-step} method using \textit{converse modalities}
  - See \textit{Modal Logic} by Blackburn et al.; more references at the end
The Difference Modality

The difference modality is a diamond \( \langle D \rangle \), where \( \langle D \rangle \phi \) means that \( \phi \) is true somewhere else.
The Difference Modality

The difference modality is a diamond \( \langle D \rangle \), where \( \langle D \rangle \phi \) means that \( \phi \) is true somewhere else.

It has a fixed interpretation in a model \( M \):

\[
M, w \models \langle D \rangle \phi \iff \exists v \neq w \ M, v \models \phi
\]
The Difference Modality

The difference modality is a diamond $\langle D \rangle$, where $\langle D \rangle \phi$ means that $\phi$ is true somewhere else.

It has a fixed interpretation in a model $M$:

$$M, w \models \langle D \rangle \phi \iff \exists v \neq w M, v \models \phi$$

- We add the difference modality to the language
- Not only to be able to axiomatise the logic
- .. but also because it is useful for reasoning about games. E.g. the core is not empty:

$$MCNE \equiv MCM \lor \langle D \rangle MCM$$
Converse Modalities

Let $\langle i^c \rangle$ denote the converse of the diamond $\langle i \rangle$:

- $\langle i \rangle \phi$: there is an outcome which is preferred by $i$ over the current one, in which $\phi$ is true
- $\langle i^c \rangle \phi$: there is an outcome over which the current outcome is preferred by $i$, in which $\phi$ is true
Converse Modalities

Let $\langle i^c \rangle$ denote the converse of the diamond $\langle i \rangle$:

- $\langle i \rangle \phi$: there is an outcome which is preferred by $i$ over the current one, in which $\phi$ is true
- $\langle i^c \rangle \phi$: there is an outcome over which the current outcome is preferred by $i$, in which $\phi$ is true

- We include converses for all the diamonds
- Convereses make the step-by-step model construction technique we are going to use possible
- Also useful for reasoning about games
9. Axiom. of Coal. Games

MCGL: Full language and explicit models

\[ \phi ::= p \mid \langle C \rangle \phi \mid \langle C^s \rangle \phi \mid \langle D \rangle \phi \mid \langle C^c \rangle \phi \mid \langle C^{sc} \rangle \phi \mid \neg \phi \mid \phi_1 \land \phi_2 \]
MCGL: Full language and explicit models

\[ \phi ::= p \mid \langle C \rangle \phi \mid \langle C^s \rangle \phi \mid \langle D \rangle \phi \mid \langle C^c \rangle \phi \mid \langle C^{sc} \rangle \phi \mid \neg \phi \mid \phi_1 \land \phi_2 \]

\[ M = (W, \{ R_C : C \in C \}, \{ R^s_C : C \in C \}, D, \{ R^c_C : C \in C \}, \{ R^{sc}_C : C \in C \}, \pi) \]

**REFL** \( \forall i \in N \, R_i \) is reflexive

**TRANS** \( \forall i \in N \, R_i \) is transitive

**COMPL** \( \forall i \in N \, R_i \) is complete

**STRICT** \( \forall i \in N \, R^s_i wu \) iff both \( R_i wu \) and not \( R_i uw \)

**DIFF** \( D = \{(w, u) : w \neq u\} \)

**INTERSECTION** \( \forall C \in C \, R_C = \bigcap_{i \in C} R_i \)

**INTERSECTION-STRCT** \( \forall C \in C \, R^s_C = \bigcap_{i \in C} R^s_i \)

**CONVERSE** \( Rwv \) iff \( R^c_vw \), for \( R \in \{ R_i, R^s_i, R_C, R^{sc}_C, D \} \)
Explicit models vs. games

Recall the interpretation in coalitional games:

\[ \Gamma, \pi, \omega \models \phi \]

- Explicit models are just another representation of \((\Gamma, \pi)\) pairs
- An axiomatisation of models will be an axiomatisation of games as well
Axioms: overview

- **Normality:** *Modus Ponens, Usub, Prop*, as well as *K* and *Nec* for all the boxes
- *T*, *4* for individual preference relations
- Axioms and rules for the difference modality
- Axioms for completeness of individual preferences
- Converse axioms
- Strictness axioms
- Intersection axioms

(see the paper p. 34 for a summary)
Axioms: difference modality

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Formula</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_1$</td>
<td>$p \rightarrow [D]⟨D⟩p$</td>
<td>$D$-rule</td>
</tr>
<tr>
<td>$D_2$</td>
<td>$◊_1 \cdots ◊_k p \rightarrow (p \lor ⟨D⟩p)$</td>
<td>symmetry</td>
</tr>
<tr>
<td>$D−rule$</td>
<td>$\vdash (p \land \neg⟨D⟩p) \rightarrow \theta \Rightarrow \vdash \theta$</td>
<td>$p$ not in $\theta$</td>
</tr>
</tbody>
</table>

Relatively standard, see Blackburn et al., *Modal Logic*. 
Axioms: completeness/totality

\begin{tabular}{|l|l|}
\hline
Trichotomy & $(p \land [i]q) \rightarrow [D](q \lor p \lor \langle i \rangle p)$ \\
\hline
\end{tabular}
Axioms: converses

\begin{tabular}{|l|l|l|}
\hline
Converse_1(\xi) & p \rightarrow [\xi] (\xi^c)p & \xi \in \Xi \\
\hline
Converse_2(\xi) & p \rightarrow [\xi^c] (\xi)p & \xi \in \Xi \\
\hline
\end{tabular}

where:

\[ \Xi = \{ C, C^s, C^c, C^{sc} \} \]
## Axioms: strictness

<table>
<thead>
<tr>
<th>Axiom</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Strict_1$</td>
<td>$p \land \langle i \rangle (q \land [i] \neg p) \rightarrow \langle i^s \rangle q$</td>
</tr>
<tr>
<td>$Strict_2$</td>
<td>$(p \land [D] \neg p \land \langle i^s \rangle q) \rightarrow \langle i \rangle (q \land \neg \langle i \rangle p)$</td>
</tr>
<tr>
<td>$Strict_3$</td>
<td>$\langle i^s \rangle p \rightarrow \langle D \rangle p$</td>
</tr>
</tbody>
</table>
### Axioms: intersection

<table>
<thead>
<tr>
<th>Intersect&lt;sub&gt;1&lt;/sub&gt;</th>
<th>$((p \land [D] \neg p) \lor \langle D \rangle (p \land [D] \neg p)) \rightarrow (\bigwedge_{i \in C} \langle i \rangle p \rightarrow \langle C \rangle p)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intersect&lt;sub&gt;2&lt;/sub&gt;</td>
<td>$((p \land [D] \neg p) \lor \langle D \rangle (p \land [D] \neg p)) \rightarrow (\bigwedge_{i \in C} \langle i^s \rangle p \rightarrow \langle C^s \rangle p)$</td>
</tr>
<tr>
<td>Intersect&lt;sub&gt;3&lt;/sub&gt;</td>
<td>$\langle C \rangle p \rightarrow \langle i \rangle p$ \hspace{1cm} $i \in C$</td>
</tr>
<tr>
<td>Intersect&lt;sub&gt;4&lt;/sub&gt;</td>
<td>$\langle C^s \rangle p \rightarrow \langle i^s \rangle p$ \hspace{1cm} $i \in C$</td>
</tr>
</tbody>
</table>
9.5 Completeness
Outline

We will use a step-by-step method:
Outline

We will use a step-by-step method:

- The result will be a submodel of the canonical model
Outline

We will use a step-by-step method:

- The result will be a submodel of the canonical model
- We will build a network, which has much of the information needed for a proper model
Outline

We will use a step-by-step method:

- The result will be a submodel of the canonical model
- We will build a network, which has much of the information needed for a proper model
- We build it step-by-step, by starting with a small network and repairing its defects by extending it
Outline

We will use a step-by-step method:

- The result will be a submodel of the canonical model
- We will build a network, which has much of the information needed for a proper model
- We build it step-by-step, by starting with a small network and repairing its defects by extending it
- The converse operators make it possible to go back and forth, and to describe a finite network using formulae
Outline

We will use a step-by-step method:

- The result will be a submodel of the canonical model
- We will build a network, which has much of the information needed for a proper model
- We build it step-by-step, by starting with a small network and repairing its defects by extending it
- The converse operators make it possible to go back and forth, and to describe a finite network using formulae
- The $D$ operator lets us
  - Define the needed model properties
  - Construct a named model by requiring that a formula of the form $p \land \neg\langle D \rangle p$ holds in a state
Outline

We will use a step-by-step method:
- The result will be a submodel of the canonical model
- We will build a network, which has much of the information needed for a proper model
- We build it step-by-step, by starting with a small network and repairing its defects by extending it
- The converse operators make it possible to go back and forth, and to describe a finite network using formulae
- The $D$ operator lets us
  - Define the needed model properties
  - Construct a named model by requiring that a formula of the form $p \land \neg[D]p$ holds in a state
Definition 9.10 (Network)

A network is a tuple

\[ \mathcal{N} = (N, E, d, r, \Lambda) \]

- \((N, E)\) is a finite, undirected, connected and acyclic graph
- \(d\) maps each edge \(\{s, t\} \in E\) to a relation in the set \(\{R_C, R^s_C, D : C \in \mathcal{C}\}\)
- \(r\) maps each edge \(\{s, t\} \in E\) to either \(s\) or \(t\)
- \(\Lambda\) labels each node in \(N\) with a finite set of formulae
We can describe a network with formulae: Let $E(s)$ denote the set of nodes adjacent to $s$, and let

\[
\langle st \rangle = \begin{cases} 
\langle i \rangle & d(\{s, t\}) = R_i \text{ and } r(\{s, t\}) = s \\
\langle ic \rangle & d(\{s, t\}) = R_i \text{ and } r(\{s, t\}) = t \\
\langle is \rangle & d(\{s, t\}) = R_i^s \text{ and } r(\{s, t\}) = s \\
\langle isc \rangle & d(\{s, t\}) = R_i^s \text{ and } r(\{s, t\}) = t \\
\langle C \rangle & d(\{s, t\}) = R_C \text{ and } r(\{s, t\}) = s \\
\langle Cc \rangle & d(\{s, t\}) = R_C \text{ and } r(\{s, t\}) = t \\
\langle Cs \rangle & d(\{s, t\}) = R_C^s \text{ and } r(\{s, t\}) = s \\
\langle Csc \rangle & d(\{s, t\}) = R_C^s \text{ and } r(\{s, t\}) = t \\
\langle D \rangle & d(\{s, t\}) = D \text{ and } r(\{s, t\}) = s \\
\langle Dc \rangle & d(\{s, t\}) = D \text{ and } r(\{s, t\}) = t \\
\end{cases}
\]

\[
\Delta(N, s) = \bigwedge \Lambda(s) \land \bigwedge_{v \in E(s)} \langle sv \rangle \Phi(N, v, s) \\
\Phi(N, t, s) = \bigwedge \Lambda(t) \land \bigwedge_{s \neq v \in E(t)} \langle tv \rangle \Phi(N, v, t)
\]

Note the role of converses for all the diamonds here!
We can describe a network with formulae:
Let $E(s)$ denote the set of nodes adjacent to $s$, and let

$$\langle i \rangle \quad d(\{s, t\}) = R_i \text{ and } r(\{s, t\}) = s$$
$$\langle i^c \rangle \quad d(\{s, t\}) = R_i \text{ and } r(\{s, t\}) = t$$
$$\langle i^s \rangle \quad d(\{s, t\}) = R_i^s \text{ and } r(\{s, t\}) = s$$
$$\langle i^{sc} \rangle \quad d(\{s, t\}) = R_i^s \text{ and } r(\{s, t\}) = t$$
$$\langle C \rangle \quad d(\{s, t\}) = R_C \text{ and } r(\{s, t\}) = s$$
$$\langle C^c \rangle \quad d(\{s, t\}) = R_C \text{ and } r(\{s, t\}) = t$$
$$\langle C^s \rangle \quad d(\{s, t\}) = R_C^s \text{ and } r(\{s, t\}) = s$$
$$\langle C^{sc} \rangle \quad d(\{s, t\}) = R_C^s \text{ and } r(\{s, t\}) = t$$
$$\langle D \rangle \quad d(\{s, t\}) = D \text{ and } r(\{s, t\}) = s$$
$$\langle D \rangle \quad d(\{s, t\}) = D \text{ and } r(\{s, t\}) = t$$

$$\Delta(\mathcal{N}, s) = \bigwedge \Lambda(s) \land \bigwedge_{v \in E(s)} \langle sv \rangle \Phi(\mathcal{N}, v, s)$$
$$\Phi(\mathcal{N}, t, s) = \bigwedge \Lambda(t) \land \bigwedge_{s \neq v \in E(t)} \langle tv \rangle \Phi(\mathcal{N}, v, t)$$

Note the role of \textbf{converses} for all the diamonds here!
We can show the following:

**Proposition**

\( \Delta(\mathcal{N}, s) \) is consistent iff \( \Delta(\mathcal{N}, t) \) is consistent, for any two nodes in any network \( \mathcal{N} \), by using the \textit{Converse}_1 and \textit{Converse}_2 axioms (when an edge is marked with anything else than a \textit{D}) and the \textit{D}_2 axiom (when an edge is marked with \textit{D}).
We can show the following:

**Proposition**

\[ \Delta(N, s) \text{ is consistent iff } \Delta(N, t) \text{ is consistent, for any two nodes in any network } N \]

by using the \text{Converse}_1 and \text{Converse}_2 axioms (when an edge is marked with anything else than a \( D \)) and the \( D_2 \) axiom (when an edge is marked with \( D \)).

**Definition 9.11 (Coherence)**

A network is coherent if \( \Delta(N, s) \) is consistent for any \( s \).
Possible **defects** in a network:

\[ D_1(s, \phi) \] where \( s \) is a node and \( \phi \) a formula, and
\( \phi \not\in \Lambda(s) \) and \( \neg \phi \not\in \Lambda(s) \)

\[ D_2(s) \] there is no formula \( \phi \) such that
\( \phi \land \neg \langle D \rangle \phi \in \Lambda(s) \)

\[ D_3(s, \langle \xi \rangle \phi) \] (\( \xi \in \{i, C, i^s, C^s, D\} \)) where \( s \) is a node and
\( \langle \xi \rangle \phi \in \Lambda(s) \) and for all \((s, t) \in E\) such that
\( d(\{s, t\}) = \text{Rel}(\xi) \) and \( r(\{s, t\}) = s \) it is the case
that \( \phi \not\in \Lambda(t) \)

\[ D_4(s, \langle \xi^c \rangle \phi) \] (\( \xi \in \{i, C, i^s, C^s\} \)) where \( s \) is a node and
\( \langle \xi^c \rangle \phi \in \Lambda(s) \) and for all \((s, t) \in E\) such that
\( d(\{s, t\}) = \text{Rel}(\xi) \) and \( r(\{s, t\}) = t \) it is the case
that \( \phi \not\in \Lambda(t) \)
Proposition

For any defect in a coherent network $\mathcal{N}$, there is a coherent network $\mathcal{N}'$ extending $\mathcal{N}$ lacking that effect.

Repairing defects: standard approach
Repairing D2-defects with the $D$-rule

$D2(s)$ there is no formula $\phi$ such that
\[ \phi \land \neg \langle D \rangle \phi \in \Lambda(s) \]

- Let $p$ be an atom not occurring in $\Delta(\mathcal{N}, s)$ (recall that we assumed there are infinitely many)
- Alternative statement of the $D$-rule:
  
  If $\Phi$ is consistent and does not contain $p$
  
  \[ \Downarrow \]
  
  \[ (p \land \neg \langle D \rangle p) \land \Phi \text{ is consistent} \]

- $\Delta(\mathcal{N}, s)$ consistent so $\Delta(\mathcal{N}, s) \land p \land \neg \langle D \rangle p$ is consistent
- Define the new network by adding $p \land \neg \langle D \rangle p$ to $\Lambda(s)$
- Clearly, it is coherent
Repairing D3- and D4-defects

\[ D3(s, \langle \xi \rangle \phi) \]  
\( (\xi \in \{i, C, i^s, C^s, D\}) \) where \( s \) is a node and  
\( \langle \xi \rangle \phi \in \Lambda(s) \) and for all \( (s, t) \in E \) such that  
d(\{s, t\}) = \text{Rel}(\xi) \) and \( r(\{s, t\}) = s \) it is the case  
that \( \phi \notin \Lambda(t) \)
Repairing D3- and D4-defects

\[ D_3(s, \langle \xi \rangle \phi) \] (\( \xi \in \{i, C, i^s, C^s, D\} \)) where s is a node and
\( \langle \xi \rangle \phi \in \Lambda(s) \) and for all \((s, t) \in E\) such that
\( d(\{s, t\}) = \text{Rel}(\xi) \) and \( r(\{s, t\}) = s \) it is the case
that \( \phi \not\in \Lambda(t) \)

We define \( \mathcal{N}' \) as follows:

- \( \mathcal{N}' = \mathcal{N} \cup \{t\} \) for some \( t \in Y \setminus \mathcal{N} \)
- \( E' = E \cup \{\{s, t\}\} \)
- \( d' = d \cup \{\{s, t\} \mapsto \text{Rel}(\xi)\} \)
- \( r' = r \cup \{\{s, t\} \mapsto s\} \)
- \( \Lambda' = \Lambda \cup \{t \mapsto \{\phi\}\} \)

where \( Y \) is a countably infinite set of “fresh” states.
9. Axiom. of Coal. Games

5. Completeness

Repairing D3- and D4-defects

\[ D3(s, \langle \xi \rangle \phi) \] (\( \xi \in \{i, C, i^s, C^s, D\} \)) where \( s \) is a node and \( \langle \xi \rangle \phi \in \Lambda(s) \) and for all \((s, t) \in E\) such that \( d(\{s, t\}) = \text{Rel}(\xi) \) and \( r(\{s, t\}) = s \) it is the case that \( \phi \notin \Lambda(t) \)

We define \( \mathcal{N}' \) as follows:

- \( \mathcal{N}' = \mathcal{N} \cup \{t\} \) for some \( t \in Y \setminus \mathcal{N} \)
- \( E' = E \cup \{\{s, t\}\} \)
- \( d' = d \cup \{\{s, t\} \mapsto \text{Rel}(\xi)\} \)
- \( r' = r \cup \{\{s, t\} \mapsto s\} \)
- \( \Lambda' = \Lambda \cup \{t \mapsto \{\phi}\} \)

where \( Y \) is a countably infinite set of “fresh” states.

We have that \( \Delta(\mathcal{N}', s) = \Delta(\mathcal{N}, s) \land \langle \xi \rangle \phi \). Since \( \langle \xi \rangle \phi \in \Lambda(s) \), it already is a conjunct of \( \Delta(\mathcal{N}, s) \). Thus, \( \mathcal{N}' \) is coherent.
Fix a consistent formula $\hat{\phi}$

We now will construct a model for it.
For every number $i$ define a network $\mathcal{N}_i = (N_i, E_i, d_i, r_i, \Lambda_i)$:

- $\mathcal{N}_0$ has a single node $y$ labelled with $\{\hat{\phi}\}$. Clearly, $\mathcal{N}_0$ is coherent.
- When $n > 0$, $\mathcal{N}_{n+1}$ is the (coherent) network obtained by repairing the next (according to some enumeration) defect, by the rules given above.
For every number $i$ define a network 
$\mathcal{N}_i = (N_i, E_i, d_i, r_i, \Lambda_i)$:

- $\mathcal{N}_0$ has a single node $y$ labelled with $\{\hat{\phi}\}$. Clearly, $\mathcal{N}_0$ is coherent.
- When $n > 0$, $\mathcal{N}_{n+1}$ is the (coherent) network obtained by repairing the next (according to some enumeration) defect, by the rules given above.

Note that:

- $\mathcal{N}_j$ extends $\mathcal{N}_i$ when $i < j$
- A repaired defect will never be reintroduced
- for any defect of $\mathcal{N}_i$ there is a $j > i$ such that $\mathcal{N}_j$ lacks that defect
Collect repairs: \( \mathcal{N} \)

Collect all repairs by defining \( \mathcal{N} = (N, E, d, r, \Lambda) \):

- \( N = \bigcup_{i \in \mathbb{N}} N_i \)
- \( E = \bigcup_{i \in \mathbb{N}} E_i \)
- \( d = \bigcup_{i \in \mathbb{N}} d_i \)
- \( r = \bigcup_{i \in \mathbb{N}} r_i \)
- \( \Lambda(s) = \bigcup \{ \Lambda_i(s) : i \in \mathbb{N}, s \in N_i \} \)
Collect repairs: $\mathcal{N}$

Collect all repairs by defining $\mathcal{N} = (N, E, d, r, \Lambda)$:

- $N = \bigcup_{i \in \mathbb{N}} N_i$
- $E = \bigcup_{i \in \mathbb{N}} E_i$
- $d = \bigcup_{i \in \mathbb{N}} d_i$
- $r = \bigcup_{i \in \mathbb{N}} r_i$
- $\Lambda(s) = \bigcup\{\Lambda_i(s) : i \in \mathbb{N}, s \in N_i\}$

Proposition

For every $s$, $\Lambda(s)$ is a maximal consistent set of formulae.

- Maximaliy: repair of $D1$ effects
- Consistency: from consistency of each $\mathcal{N}_i$
The Model

Define the model $M$ (for $\hat{\phi}$) by restricting the canonical model for MCGL to the MCSs that appear in $\mathcal{N}$, i.e. to

$$W = \{ \Lambda(s) : s \in N \}$$
The Model

Define the model $M$ (for $\hat{\phi}$) by restricting the canonical model for MCGL to the MCSs that appear in $\mathcal{N}$, i.e. to

$$W = \{ \Lambda(s) : s \in N \}$$

(remove the other states, restrict the relations and valuation function accordingly)
9. Axiom. of Coal. Games

5. Completeness

Truth Lemma

Proposition

\[ M, \Gamma \models \psi \iff \psi \in \Gamma \]

for any \( \Gamma \in W \) and any \( \psi \)
9. Axiom. of Coal. Games

5. Completeness

Truth Lemma

Proposition

\[ M, \Gamma \models \psi \iff \psi \in \Gamma \]

for any \( \Gamma \in W \) and any \( \psi \)

- First: show that we don’t throw away too much, that for any diamond \( \Diamond \) we have that \( \Gamma \in W \) whenever \( \Diamond \psi \in \Gamma \), there is a \( \Delta \in W \) such that \( \psi \in \text{Delta} \) and \( \Gamma, \Delta \) are related by the canonical relation. Easily shown by construction.
- Then: induction on \( \phi \)
It remains to be shown that $M$ has all the properties we required.
It remains to be shown that $M$ has all the properties we required.

**REFL** $\forall i \in N \ R_i$ is reflexive

**TRANS** $\forall i \in N \ R_i$ is transitive

**COMPL** $\forall i \in N \ R_i$ is complete

**STRICT** $\forall i \in N \ R^s_iwu$ iff both $R_iwu$ and not $R_iuw$

**DIFF** $D = \{(w, u) : w \neq u\}$

**INTERSECTION** $\forall C \in C \ R_C = \bigcap_{i \in C} R_i$

**INTERSECTION-STRICT** $\forall C \in C \ R^s_C = \bigcap_{i \in C} R^s_i$

**CONVERSE** $R^c vw$ iff $R^c vw$, for $R \in \{R_i, R^s_i, R_C, R^s_C, D\}$
$M$ is named

Because we removed $D^2$-defects, for every state $w$ there exists a formula $\phi_w$ such that

$$M, w \models \phi_w \land \neg \langle D \rangle \phi_w$$
9. Axiom. of Coal. Games

5. Completeness

$M$ is named

Because we removed $D2$-defects, for every state $w$ there exists a formula $\phi_w$ such that

$$M, w \models \phi_w \land \neg \langle D \rangle \phi_w$$

But from DIFF it follows that $\phi_w$ is uniquely true at $w$, that

$$M, w \models \phi_w$$

and for any $u \neq w$

$$M, u \not\models \phi_w$$
ASSUME that $w \neq u$, $\neg R_iuw$ and $\neg R_iwu$
COMPL

- Assume that \( w \neq u, \neg R_iwu \) and \( \neg R_iwu \).
- We have that \( M, u \not\models \langle i^c \rangle \phi_w \): otherwise
  - there is a \( v \) s.t. \( R^e_i uv \) and \( M, v \models \phi_w \)
  - thus \( v = w \), and \( R^e_i uw \)
  - by CONVERSE: \( R_iwu \) – a contradiction
9. Axiom. of Coal. Games

5. Completeness

COMPL

- Assume that $w \neq u$, $\neg R_iuw$ and $\neg R_iwu$
- We have that $M, u \not\models \langle i^c \rangle \phi_w$: otherwise
  - there is a $v$ s.t. $R_i^c uv$ and $M, v \models \phi_w$
  - thus $v = w$, and $R_i^c uw$
  - by CONVERSE: $R_iwu$ – a contradiction
- Thus $M, u \not\models (\phi_w \vee \langle i^c \rangle \phi_w \vee \langle i \rangle \phi_w)$
9. Axiom. of Coal. Games

5. Completeness

COMPL

- Assume that \( w \neq u, \neg R_{i}uw \) and \( \neg R_{i}wu \)
- We have that \( M, u \nvDash \langle i^{c} \rangle \phi_{w} \): otherwise
  - there is a \( v \) s.t. \( R_{c}iuv \) and \( M, v \vDash \phi_{w} \)
  - thus \( v = w \), and \( R_{c}iuw \)
  - by CONVERSE: \( R_{i}wu \) – a contradiction
- Thus \( M, u \nvDash (\phi_{w} \lor \langle i^{c} \rangle \phi_{w} \lor \langle i \rangle \phi_{w}) \)
- And since \( w \neq u \), \( M, w \nvDash [D]((\phi_{w} \lor \langle i^{c} \rangle \phi_{w} \lor \langle i \rangle \phi_{w}) \)
9. Axiom. of Coal. Games

5. Completeness

COMPL

- Assume that \( w \neq u, \neg R_iuw \) and \( \neg R_iwu \)
- We have that \( M, u \not\models \langle i^c \rangle \phi_w \): otherwise
  - there is a \( v \) s.t. \( R^c_iuv \) and \( M, v \models \phi_w \)
  - thus \( v = w \), and \( R^c_iuw \)
  - by CONVERSE: \( R_iwu \) – a contradiction
- Thus \( M, u \not\models (\phi_w \lor \langle i^c \rangle \phi_w \lor \langle i \rangle \phi_w) \)
- And since \( w \neq u, M, w \not\models [D]((\phi_w \lor \langle i^c \rangle \phi_w \lor \langle i \rangle \phi_w) \)
- Also: \( M, w \models (\phi_w \land [i] \langle i^c \rangle \phi_w) \)
COMPL

- Assume that $w \neq u$, $\neg R_i uw$ and $\neg R_i wu$
- We have that $M, u \not\models \langle i^c \rangle \phi_w$: otherwise
  - there is a $v$ s.t. $R_c i uv$ and $M, v \models \phi_w$
  - thus $v = w$, and $R_c i uw$
  - by CONVERSE: $R_i wu$ – a contradiction

- Thus $M, u \not\models (\phi_w \lor \langle i^c \rangle \phi_w \lor \langle i \rangle \phi_w)$
- And since $w \neq u$, $M, w \not\models [D]((\phi_w \lor \langle i^c \rangle \phi_w \lor \langle i \rangle \phi_w)$
- Also: $M, w \models (\phi_w \land [i]\langle i^c \rangle \phi_w)$
- Contradicts the Trichotomy axiom
9. Axiom. of Coal. Games

6. References

_A Course in Game Theory._

Reasoning about coalitional games.
To appear in _Artificial Intelligence_, 2008.

_Extended Modal Logic._

_Modal Logic._
Thank you for your attention!