Knowledge and Cooperation in Infinite Games

LINT Workshop, Paris, 22. Jul. - 24. Jul. 2010

Bernd Puchala

RWTH Aachen University

Epistemology

Epistemology

- Reasoning about knowledge originates in Philosophy

Epistemology

- Reasoning about knowledge originates in Philosophy
- Epistemology analyzes the nature of knowledge

Epistemology

- Reasoning about knowledge originates in Philosophy
- Epistemology analyzes the nature of knowledge
- and tries to answer questions like

Epistemology

- Reasoning about knowledge originates in Philosophy
- Epistemology analyzes the nature of knowledge
- and tries to answer questions like
    - What is knowledge?

Epistemology

- Reasoning about knowledge originates in Philosophy
- Epistemology analyzes the nature of knowledge
- and tries to answer questions like
  - What is knowledge?
  - How is knowledge acquired?

Epistemology

- Reasoning about knowledge originates in Philosophy
- Epistemology analyzes the nature of knowledge
- and tries to answer questions like
  - What is knowledge?
  - How is knowledge acquired?
  - What do people know?

Epistemology

- Reasoning about knowledge originates in Philosophy
- Epistemology analyzes the nature of knowledge
- and tries to answer questions like
  - What is knowledge?
  - How is knowledge acquired?
  - What do people know?
  - How do we know what we know?

Epistemic Logic

Epistemic Logic

- Idea: Put the analysis of these questions on a formal logical ground

Epistemic Logic

- Idea: Put the analysis of these questions on a formal logical ground
- Jaakko Hintikka, 1962: *Knowledge and Belief: An Introduction to the Logic of the Two Notions*

Epistemic Logic

- Idea: Put the analysis of these questions on a formal logical ground
- Jaakko Hintikka, 1962: *Knowledge and Belief: An Introduction to the Logic of the Two Notions*
- Hintikka tried to capture inherent properties of knowledge by formal logical rules

Epistemic Logic

- Idea: Put the analysis of these questions on a formal logical ground
- Jaakko Hintikka, 1962: *Knowledge and Belief: An Introduction to the Logic of the Two Notions*
- Hintikka tried to capture inherent properties of knowledge by formal logical rules
- and used Modal Logic with Possible World Semantics

Epistemic Logic

- Idea: Put the analysis of these questions on a formal logical ground
- Jaakko Hintikka, 1962: *Knowledge and Belief: An Introduction to the Logic of the Two Notions*
- Hintikka tried to capture inherent properties of knowledge by formal logical rules
- and used Modal Logic with Possible World Semantics
- Modal Logic already used by Aristotle

Epistemic Logic

- Idea: Put the analysis of these questions on a formal logical ground
- Jaakko Hintikka, 1962: *Knowledge and Belief: An Introduction to the Logic of the Two Notions*
- Hintikka tried to capture inherent properties of knowledge by formal logical rules
- and used Modal Logic with Possible World Semantics
- Modal Logic already used by Aristotle
- Possible Worlds Semantics developed by Carnap, Hintikka, Kripke, . . .

Epistemic Logic

- Idea: Put the analysis of these questions on a formal logical ground
- Jaakko Hintikka, 1962: *Knowledge and Belief: An Introduction to the Logic of the Two Notions*
- Hintikka tried to capture inherent properties of knowledge by formal logical rules
- and used Modal Logic with Possible World Semantics
- Modal Logic already used by Aristotle
- Possible Worlds Semantics developed by Carnap, Hintikka, Kripke, . . .
- Current form: Kripke structures

Epistemic Logic

Epistemic Logic

A Kripke-structure has the form

$$\mathcal{K} = (V, \mathrm{Prop}, (E_i)_{i \in I})$$

where

Epistemic Logic

A Kripke-structure has the form

$$\mathcal{K} = (V, \mathrm{Prop}, (E_i)_{i \in I})$$

where

- V is a (finite) set (of possible worlds)

Epistemic Logic

A Kripke-structure has the form

$$\mathcal{K} = (V, \mathrm{Prop}, (E_i)_{i \in I})$$

where

- V is a (finite) set (of possible worlds)
- Prop is a set of unary relations (atomic propositions)

Epistemic Logic

A Kripke-structure has the form

$$\mathcal{K} = (V, \mathrm{Prop}, (E_i)_{i \in I})$$

where

- V is a (finite) set (of possible worlds)
- Prop is a set of unary relations (atomic propositions)
- each E_i is a binary relation (alternative relation)

Epistemic Logic

A Kripke-structure has the form

$$\mathcal{K} = (V, \mathrm{Prop}, (E_i)_{i \in I})$$

where

- V is a (finite) set (of possible worlds)
- Prop is a set of unary relations (atomic propositions)
- each E_i is a binary relation (alternative relation)

Epistemic Logic is defined by

Epistemic Logic

A Kripke-structure has the form

$$\mathcal{K} = (V, \mathrm{Prop}, (E_i)_{i \in I})$$

where

- V is a (finite) set (of possible worlds)
- Prop is a set of unary relations (atomic propositions)
- each E_i is a binary relation (alternative relation)

Epistemic Logic is defined by

$$\varphi ::= P \mid \varphi \wedge \varphi \mid \neg\varphi \mid K_i\varphi$$

Epistemic Logic

A Kripke-structure has the form

$$\mathcal{K} = (V, \mathrm{Prop}, (E_i)_{i \in I})$$

where

- V is a (finite) set (of possible worlds)
- Prop is a set of unary relations (atomic propositions)
- each E_i is a binary relation (alternative relation)

Epistemic Logic is defined by

$$\varphi ::= P \,|\, \varphi \wedge \varphi \,|\, \neg\varphi \,|\, K_i\varphi$$

- $\mathcal{K}, v \models P$ iff $v \in P$

Epistemic Logic

A Kripke-structure has the form

$$\mathcal{K} = (V, \mathrm{Prop}, (E_i)_{i \in I})$$

where

- V is a (finite) set (of possible worlds)
- Prop is a set of unary relations (atomic propositions)
- each E_i is a binary relation (alternative relation)

Epistemic Logic is defined by

$$\varphi ::= P \mid \varphi \wedge \varphi \mid \neg\varphi \mid K_i\varphi$$

- $\mathcal{K}, v \models P$ iff $v \in P$
- $\mathcal{K}, v \models K_i\varphi$ iff $\mathcal{K}, w \models \varphi$ for all $w \in V$ with $(v, w) \in E_i$

Epistemic Logic

Epistemic Logic

- This formal treatment of knowledge has many applications:

Epistemic Logic

- This formal treatment of knowledge has many applications:
  - Artificial Intelligence

Epistemic Logic

- This formal treatment of knowledge has many applications:
  - Artificial Intelligence
    A robot should not only complete his task but he should also *know* when his task is completed

Epistemic Logic

- This formal treatment of knowledge has many applications:
  - Artificial Intelligence
    A robot should not only complete his task but he should also *know* when his task is completed
  - Synthesis of systems with partial obervsation

Epistemic Logic

- This formal treatment of knowledge has many applications:
  - Artificial Intelligence
    A robot should not only complete his task but he should also *know* when his task is completed
  - Synthesis of systems with partial obervsation
    A safety critical action should only be performed by a system, when the controller *knows* that the current state of the system ensures a safe execution

Epistemic Logic

- This formal treatment of knowledge has many applications:
  - Artificial Intelligence
    A robot should not only complete his task but he should also *know* when his task is completed
  - Synthesis of systems with partial obervsation
    A safety critical action should only be performed by a system, when the controller *knows* that the current state of the system ensures a safe execution
    Security protocols often involve requirements like “no component of the system will ever *know* the value of any internal variable of some other component”

Epistemic Logic

- This formal treatment of knowledge has many applications:
  - Artificial Intelligence
    A robot should not only complete his task but he should also *know* when his task is completed
  - Synthesis of systems with partial obervsation
    A safety critical action should only be performed by a system, when the controller *knows* that the current state of the system ensures a safe execution
    Security protocols often involve requirements like "no component of the system will ever *know* the value of any internal variable of some other component"
  - Linguistics

Epistemic Logic

- This formal treatment of knowledge has many applications:
  - Artificial Intelligence
    A robot should not only complete his task but he should also *know* when his task is completed
  - Synthesis of systems with partial obervsation
    A safety critical action should only be performed by a system, when the controller *knows* that the current state of the system ensures a safe execution
    Security protocols often involve requirements like "no component of the system will ever *know* the value of any internal variable of some other component"
  - Linguistics
  - Economics

Epistemic Logic

- This formal treatment of knowledge has many applications:
  - Artificial Intelligence
    A robot should not only complete his task but he should also *know* when his task is completed
  - **Synthesis of systems with partial obervsation**
    A safety critical action should only be performed by a system, when the controller *knows* that the current state of the system ensures a safe execution
    Security protocols often involve requirements like "no component of the system will ever *know* the value of any internal variable of some other component"
  - Linguistics
  - Economics

Epistemic Logic

It is often assumed that the alternative relations E_i are equivalence relations $\sim_i$.

Epistemic Logic

It is often assumed that the alternative relations E_i are equivalence relations $\sim_i$.
This has far reaching consequences on the logical system!

Epistemic Logic

It is often assumed that the alternative relations E_i are equivalence relations $\sim_i$.
This has far reaching consequences on the logical system!

- $K_i\varphi \rightarrow \varphi$ (Knowledge Axiom)

Epistemic Logic

It is often assumed that the alternative relations E_i are equivalence relations $\sim_i$.

This has far reaching consequences on the logical system!

- $K_i\varphi \rightarrow \varphi$ (Knowledge Axiom)
- $K_i\varphi \rightarrow K_iK_i\varphi$ (Positive Introspection Axiom)

Epistemic Logic

It is often assumed that the alternative relations E_i are equivalence relations $\sim_i$.
This has far reaching consequences on the logical system!

- $K_i\varphi \rightarrow \varphi$ (Knowledge Axiom)
- $K_i\varphi \rightarrow K_iK_i\varphi$ (Positive Introspection Axiom)
- $\neg K_i\varphi \rightarrow K_i\neg K_i\varphi$ (Negative Introspection Axiom)

Epistemic Logic

It is often assumed that the alternative relations E_i are equivalence relations $\sim_i$.
This has far reaching consequences on the logical system!

- $K_i\varphi \to \varphi$ (Knowledge Axiom)
- $K_i\varphi \to K_iK_i\varphi$ (Positive Introspection Axiom)
- $\neg K_i\varphi \to K_i\neg K_i\varphi$ (Negative Introspection Axiom)

- Positive introspection and (even more) negative introspection are highly controversial among philosophers!

Epistemic Logic

It is often assumed that the alternative relations E_i are equivalence relations $\sim_i$.
This has far reaching consequences on the logical system!

- $K_i\varphi \to \varphi$ (Knowledge Axiom)
- $K_i\varphi \to K_iK_i\varphi$ (Positive Introspection Axiom)
- $\neg K_i\varphi \to K_i\neg K_i\varphi$ (Negative Introspection Axiom)

- Positive introspection and (even more) negative introspection are highly controversial among philosophers!
- Also questionable in some applications

Epistemic Logic

It is often assumed that the alternative relations E_i are equivalence relations $\sim_i$.
This has far reaching consequences on the logical system!

- $K_i\varphi \rightarrow \varphi$ (Knowledge Axiom)
- $K_i\varphi \rightarrow K_iK_i\varphi$ (Positive Introspection Axiom)
- $\neg K_i\varphi \rightarrow K_i\neg K_i\varphi$ (Negative Introspection Axiom)

- Positive introspection and (even more) negative introspection are highly controversial among philosophers!
- Also questionable in some applications
- For our concerns justified!

Epistemic Logic

- The knowledge of a component (agent) i of a computing system is defined via the information that agent i has about the system

Epistemic Logic

- The knowledge of a component (agent) i of a computing system is defined via the information that agent i has about the system
- $u \sim_i v$ means that in state u and in state v, agent i has exactly the same information, that means, u and v are *indistinguishable* for agent i

Epistemic Logic

- The knowledge of a component (agent) i of a computing system is defined via the information that agent i has about the system
- $u \sim_i v$ means that in state u and in state v, agent i has exactly the same information, that means, u and v are *indistinguishable* for agent i
- Agent i *knows* a fact φ about the system, if φ holds in all states of the system which agent i cannot distinguish from the current state

Epistemic Logic

- The knowledge of a component (agent) i of a computing system is defined via the information that agent i has about the system
- $u \sim_i v$ means that in state u and in state v, agent i has exactly the same information, that means, u and v are *indistinguishable* for agent i
- Agent i *knows* a fact φ about the system, if φ holds in all states of the system which agent i cannot distinguish from the current state
- Computing systems evolve over time
- So does the knowledge of the agents about the system (Information flow!)

Epistemic Logic

- The knowledge of a component (agent) i of a computing system is defined via the information that agent i has about the system
- $u \sim_i v$ means that in state u and in state v, agent i has exactly the same information, that means, u and v are *indistinguishable* for agent i
- Agent i *knows* a fact φ about the system, if φ holds in all states of the system which agent i cannot distinguish from the current state
- Computing systems evolve over time
- So does the knowledge of the agents about the system (Information flow!)
- Consider logics which allow to express temporal statements about the knowledge of the agents

Epistemic Temporal Logic

Epistemic Temporal Logic

A *multi-agent system* has the form

$$\mathcal{E} = (R, \text{Prop}, \zeta, (\sim_i)_{i \in \underline{n}})$$

where

Epistemic Temporal Logic

A *multi-agent system* has the form

$$\mathcal{E} = (R, \mathrm{Prop}, \zeta, (\sim_i)_{i \in \underline{n}})$$

where

- R is a set of *runs*
- Prop is a set of atomic propositions
- $\zeta : R \times \mathbb{N} \rightarrow 2^{\mathrm{Prop}}$ is the propositional labelling
- each $\sim_i$ is an equivalence relation on $R \times \mathbb{N}$

Epistemic Temporal Logic

A *multi-agent system* has the form

$$\mathcal{E} = (R, \mathrm{Prop}, \zeta, (\sim_i)_{i \in \underline{n}})$$

where

- R is a set of *runs*
- Prop is a set of atomic propositions
- $\zeta : R \times \mathbb{N} \to 2^{\mathrm{Prop}}$ is the propositional labelling
- each $\sim_i$ is an equivalence relation on $R \times \mathbb{N}$

- The *knowledge* of agent i at some point (π, n) is given by $\sim_i$
- Equivalent points are indistinguishable for agent i, i.e., if $(\pi, n) \sim_i (\rho, m)$ then at point (π, n) and (ρ, m), agent i has exactly the same information, so he cannot distinguish one situation from the other

Epistemic Temporal Logic

Epistemic Temporal Logic

Epistemic Temporal Logic ETL is defined by

$$\varphi ::= P \mid \varphi \wedge \varphi \mid \neg\varphi \mid X\varphi \mid \varphi U \varphi \mid K_i\varphi \mid$$

where $P \in \mathrm{Prop}$ and $i \in \{1, \ldots, n\}$

Epistemic Temporal Logic

Epistemic Temporal Logic ETL is defined by

$$\varphi ::= P \mid \varphi \wedge \varphi \mid \neg\varphi \mid X\varphi \mid \varphi U \varphi \mid K_i\varphi \mid$$

where $P \in \mathrm{Prop}$ and $i \in \{1, \ldots, n\}$

$\mathcal{E}, (\pi, n) \models$

Epistemic Temporal Logic

Epistemic Temporal Logic ETL is defined by

$$\varphi ::= P \mid \varphi \wedge \varphi \mid \neg\varphi \mid X\varphi \mid \varphi U \varphi \mid K_i\varphi \mid$$

where $P \in \mathrm{Prop}$ and $i \in \{1, \ldots, n\}$

$\mathcal{E}, (\pi, n) \models$

- P iff $P \in \zeta(\pi, n)$

Epistemic Temporal Logic

Epistemic Temporal Logic ETL is defined by

$$\varphi ::= P \,|\, \varphi \wedge \varphi \,|\, \neg\varphi \,|\, X\varphi \,|\, \varphi U \varphi \,|\, K_i\varphi \,|$$

where $P \in \mathrm{Prop}$ and $i \in \{1, \ldots, n\}$

$\mathcal{E}, (\pi, n) \models$

- P iff $P \in \zeta(\pi, n)$
- $X\varphi$ iff $\mathcal{E}, (\pi, n+1) \models \varphi$

Epistemic Temporal Logic

Epistemic Temporal Logic ETL is defined by

$$\varphi ::= P \,|\, \varphi \wedge \varphi \,|\, \neg\varphi \,|\, X\varphi \,|\, \varphi U \varphi \,|\, K_i\varphi \,|$$

where $P \in \mathrm{Prop}$ and $i \in \{1, \ldots, n\}$

$\mathcal{E}, (\pi, n) \models$

- P iff $P \in \zeta(\pi, n)$
- $X\varphi$ iff $\mathcal{E}, (\pi, n+1) \models \varphi$
- $\varphi U \psi$ iff there is some $m \geq n$ such that $\mathcal{E}, (\pi, m) \models \psi$ and $\mathcal{E}, (\pi, k) \models \varphi$ for all $n \leq k < m$

Epistemic Temporal Logic

Epistemic Temporal Logic ETL is defined by

$$\varphi ::= P \,|\, \varphi \wedge \varphi \,|\, \neg\varphi \,|\, X\varphi \,|\, \varphi U \varphi \,|\, K_i\varphi \,|$$

where $P \in \mathrm{Prop}$ and $i \in \{1, \ldots, n\}$

$\mathcal{E}, (\pi, n) \models$

- P iff $P \in \zeta(\pi, n)$
- $X\varphi$ iff $\mathcal{E}, (\pi, n+1) \models \varphi$
- $\varphi U \psi$ iff there is some $m \geq n$ such that $\mathcal{E}, (\pi, m) \models \psi$ and $\mathcal{E}, (\pi, k) \models \varphi$ for all $n \leq k < m$
- $K_i\varphi$ iff $\mathcal{E}, (\rho, m) \models \varphi$ for all $(\rho, m) \sim_i (\pi, n)$

Epistemic Temporal Logic

Epistemic Temporal Logic ETL is defined by

$$\varphi ::= P \mid \varphi \wedge \varphi \mid \neg\varphi \mid X\varphi \mid \varphi U \varphi \mid K_i\varphi \mid$$

where $P \in \mathrm{Prop}$ and $i \in \{1, \ldots, n\}$

$\mathcal{E}, (\pi, n) \models$

- P iff $P \in \zeta(\pi, n)$
- $X\varphi$ iff $\mathcal{E}, (\pi, n+1) \models \varphi$
- $\varphi U \psi$ iff there is some $m \geq n$ such that $\mathcal{E}, (\pi, m) \models \psi$ and $\mathcal{E}, (\pi, k) \models \varphi$ for all $n \leq k < m$
- $K_i\varphi$ iff $\mathcal{E}, (\rho, m) \models \varphi$ for all $(\rho, m) \sim_i (\pi, n)$

$\mathcal{E} \models \varphi$ iff $\mathcal{E}, (\pi, 0) \models \varphi$ for all $\pi \in R$

Epistemic Temporal Logic

Epistemic Temporal Logic ETL+C with common knowledge is defined as ETL with the additional rule $\varphi ::= C_B\varphi$, where $B \subseteq \{1, \ldots, n\}$

Epistemic Temporal Logic

Epistemic Temporal Logic ETL+C with common knowledge is defined as ETL with the additional rule $\varphi ::= C_B\varphi$, where $B \subseteq \{1, \ldots, n\}$

$\mathcal{E}, (\pi, n) \models \mathbf{C_B}\varphi$ iff for all (ρ, m) with $(\pi, n) \sim_B (\rho, m)$ we have $\mathcal{E}, (\rho, m) \models \varphi$

where $\sim_B$ is the transitive closure of $\bigcup_{i \in B} \sim_i$.

Epistemic Temporal Logic

Epistemic Temporal Logic ETL+C with common knowledge is defined as ETL with the additional rule $\varphi ::= C_B\varphi$, where $B \subseteq \{1, \dots, n\}$

$\mathcal{E}, (\pi, n) \models \mathbf{C_B}\varphi$ iff for all (ρ, m) with $(\pi, n) \sim_B (\rho, m)$ we have $\mathcal{E}, (\rho, m) \models \varphi$

where $\sim_B$ is the transitive closure of $\bigcup_{i \in B} \sim_i$.

Common Knowledge:
$\mathcal{E}, (\pi, n) \models C_B\varphi$ iff
for all $i_1, \dots, i_k \in B$ we have $\mathcal{E}, (\pi, n) \models K_{i_1} \dots K_{i_n}\varphi$

Epistemic Temporal Logic

Epistemic Temporal Logic ETL+C with common knowledge is defined as ETL with the additional rule $\varphi ::= C_B\varphi$, where $B \subseteq \{1, \dots, n\}$

$\mathcal{E}, (\pi, n) \models \mathbf{C_B}\varphi$ iff for all (ρ, m) with $(\pi, n) \sim_B (\rho, m)$ we have $\mathcal{E}, (\rho, m) \models \varphi$

where $\sim_B$ is the transitive closure of $\bigcup_{i \in B} \sim_i$.

Common Knowledge:
$\mathcal{E}, (\pi, n) \models C_B\varphi$ iff
for all $i_1, \dots, i_k \in B$ we have $\mathcal{E}, (\pi, n) \models K_{i_1} \dots K_{i_n}\varphi$

- $\mathcal{E}, (\pi, n) \models K_i\varphi$ for all $i \in B$: Everyone in B knows φ

Epistemic Temporal Logic

Epistemic Temporal Logic ETL+C with common knowledge is defined as ETL with the additional rule $\varphi ::= C_B\varphi$, where $B \subseteq \{1,\ldots,n\}$

$\mathcal{E},(\pi,n) \models \mathbf{C_B}\varphi$ iff for all (ρ,m) with $(\pi,n) \sim_B (\rho,m)$ we have $\mathcal{E},(\rho,m) \models \varphi$

where $\sim_B$ is the transitive closure of $\bigcup_{i\in B} \sim_i$.

Common Knowledge:
$\mathcal{E},(\pi,n) \models C_B\varphi$ iff
for all $i_1,\ldots,i_k \in B$ we have $\mathcal{E},(\pi,n) \models K_{i_1}\ldots K_{i_n}\varphi$

- $\mathcal{E},(\pi,n) \models K_i\varphi$ for all $i \in B$: Everyone in B knows φ
- $\mathcal{E},(\pi,n) \models K_iK_j\varphi$ for all $i,j \in B$: Everyone in B knows that everyone in B knows φ

Epistemic Temporal Logic

Epistemic Temporal Logic ETL+C with common knowledge is defined as ETL with the additional rule $\varphi ::= C_B\varphi$, where $B \subseteq \{1, \ldots, n\}$

$\mathcal{E}, (\pi, n) \models \mathbf{C_B}\varphi$ iff for all (ρ, m) with $(\pi, n) \sim_B (\rho, m)$ we have $\mathcal{E}, (\rho, m) \models \varphi$

where $\sim_B$ is the transitive closure of $\bigcup_{i \in B} \sim_i$.

Common Knowledge:
$\mathcal{E}, (\pi, n) \models C_B\varphi$ iff
for all $i_1, \ldots, i_k \in B$ we have $\mathcal{E}, (\pi, n) \models K_{i_1} \ldots K_{i_n}\varphi$

- $\mathcal{E}, (\pi, n) \models K_i\varphi$ for all $i \in B$: Everyone in B knows φ
- $\mathcal{E}, (\pi, n) \models K_iK_j\varphi$ for all $i, j \in B$: Everyone in B knows that everyone in B knows φ
- ...

Epistemic Temporal Logic

Epistemic Temporal Logic

We are mainly interested in nonterminating finite state systems with partial observation, so we consider systems which are generated by finite systems where the agents have uncertainties about the states and actions

Epistemic Temporal Logic

We are mainly interested in nonterminating finite state systems with partial observation, so we consider systems which are generated by finite systems where the agents have uncertainties about the states and actions

A *finite multi-agent system* has the form

$$\mathcal{E} = (V, \mathrm{Prop}, \Delta, (\sim_i^V)_{i \in \underline{n}}, (\sim_i^A)_{i \in \underline{n}})$$

where

Epistemic Temporal Logic

We are mainly interested in nonterminating finite state systems with partial observation, so we consider systems which are generated by finite systems where the agents have uncertainties about the states and actions

A *finite multi-agent system* has the form

$$\mathcal{E} = (V, \mathrm{Prop}, \Delta, (\sim_i^V)_{i\in\underline{n}}, (\sim_i^A)_{i\in\underline{n}})$$

where

- V is a finite set of states
- Prop is a finite set of atomic propositions
- $\Delta \subseteq V \times A \times V$ is the move relation
- $\sim_i^V$ and $\sim_i^A$ are equivalence relations on V and A respectively

Epistemic Temporal Logic

We are mainly interested in nonterminating finite state systems with partial observation, so we consider systems which are generated by finite systems where the agents have uncertainties about the states and actions

A *finite multi-agent system* has the form

$$\mathcal{E} = (V, \mathrm{Prop}, \Delta, (\sim_i^V)_{i \in \underline{n}}, (\sim_i^A)_{i \in \underline{n}})$$

where

- V is a finite set of states
- Prop is a finite set of atomic propositions
- $\Delta \subseteq V \times A \times V$ is the move relation
- $\sim_i^V$ and $\sim_i^A$ are equivalence relations on V and A respectively
- *Run:* infinite sequence $\pi = v_0 a_1 v_1 \ldots \in V(AV)^\omega$ such that $(v_i, a_{i+1}, v_{i+1}) \in \Delta$ for each $i < \omega$

Epistemic Temporal Logic

There are many possibilities to define $\sim_i$ from $\sim_i^V$ and $\sim_i^A$

Epistemic Temporal Logic

There are many possibilities to define $\sim_i$ from $\sim_i^V$ and $\sim_i^A$

Let $\pi = v_0 a_1 v_1 \ldots$ and $\rho = w_0 b_1 w_1 \ldots$

Epistemic Temporal Logic

There are many possibilities to define $\sim_i$ from $\sim_i^V$ and $\sim_i^A$

Let $\pi = v_0 a_1 v_1 \ldots$ and $\rho = w_0 b_1 w_1 \ldots$

Here we focus on synchronous perfect recall:

$$(\pi, n) \;\sim_i^*\; (\rho, m) \;\; :\Longleftrightarrow \;\; m = n \text{ and } a_j \sim_i^A b_j,\; v_j \sim_i^V w_j$$

Epistemic Temporal Logic

There are many possibilities to define $\sim_i$ from $\sim_i^V$ and $\sim_i^A$

Let $\pi = v_0 a_1 v_1 \ldots$ and $\rho = w_0 b_1 w_1 \ldots$

Here we focus on synchronous perfect recall:

$$(\pi, n) \;\sim_i^*\; (\rho, m) \;\;:\Longleftrightarrow\;\; m = n \text{ and } a_j \sim_i^A b_j,\; v_j \sim_i^V w_j$$

We also consider an instance of asynchronous perfect recall:

$$(\pi, n) \overleftarrow{\sim}_i^* (\rho, m) \;:\Longleftrightarrow\; (\overleftarrow{\pi}, n) \sim_i^* (\overleftarrow{\rho}, m)$$

where $\overleftarrow{\pi}$ is obtained from π by contracting each maximal sequence $v_r a_{r+1} v_{r+1} \ldots a_s v_s$ with $v_j \sim_i^V v_{j+1}$ to v_r

Epistemic Temporal Logic

Observational:

Epistemic Temporal Logic

Observational:

$$(\pi, n) \ \sim_i \ (\rho, m) \ \ :\Longleftrightarrow \ \ v_n \sim_i^V w_m$$

Epistemic Temporal Logic

Observational:

$$(\pi, n) \ \sim_i \ (\rho, m) \ \ :\Longleftrightarrow \ \ v_n \sim_i^V w_m$$

Clock:

Epistemic Temporal Logic

Observational:

$$(\pi, n) \;\sim_i\; (\rho, m) \;\;:\Longleftrightarrow\;\; v_n \sim_i^V w_m$$

Clock:

$$(\pi, n) \;\sim_i\; (\rho, m) \;\;:\Longleftrightarrow\;\; n = m \text{ and } v_n \sim_i^V w_m$$

Epistemic Temporal Logic

Using this formalism, we want to check finite systems for epistemic properties:

Epistemic Temporal Logic

Using this formalism, we want to check finite systems for epistemic properties:
Given a finite system $\mathcal{E} = (V, \mathrm{Prop}, \Delta, (\sim_i^V)_{i \in \underline{n}}, (\sim_i^A)_{i \in \underline{n}})$, a state v and a formula $\varphi \in \mathrm{ETL}$

Epistemic Temporal Logic

Using this formalism, we want to check finite systems for epistemic properties:
Given a finite system $\mathcal{E} = (V, \mathrm{Prop}, \Delta, (\sim_i^V)_{i \in \underline{n}}, (\sim_i^A)_{i \in \underline{n}})$, a state v and a formula $\varphi \in \mathrm{ETL}$

$$\mathcal{E}, v \models \varphi?$$

Epistemic Temporal Logic

Using this formalism, we want to check finite systems for epistemic properties:
Given a finite system $\mathcal{E} = (V, \mathrm{Prop}, \Delta, (\sim_i^V)_{i \in \underline{n}}, (\sim_i^A)_{i \in \underline{n}})$, a state v and a formula $\varphi \in \mathrm{ETL}$

$$\mathcal{E}, v \models \varphi?$$

Model-Checking Knowledge and Time!

Epistemic Temporal Logic

Using this formalism, we want to check finite systems for epistemic properties:
Given a finite system $\mathcal{E} = (V, \mathrm{Prop}, \Delta, (\sim_i^V)_{i\in\underline{n}}, (\sim_i^A)_{i\in\underline{n}})$, a state v and a formula $\varphi \in \mathrm{ETL}$

$$\mathcal{E}, v \models \varphi?$$

Model-Checking Knowledge and Time!

$\mathcal{E}, v \models \varphi$ is defined as $\mathcal{R}(\mathcal{E}) \models \varphi$
where $\mathcal{R}(\mathcal{E}) = (R, \mathrm{Prop}, \zeta, (\sim_i)_{i\in\underline{n}})$ is the unravelling of $\mathcal{E}$ from v

Epistemic Temporal Logic

Using this formalism, we want to check finite systems for epistemic properties:
Given a finite system $\mathcal{E} = (V, \mathrm{Prop}, \Delta, (\sim_i^V)_{i \in \underline{n}}, (\sim_i^A)_{i \in \underline{n}})$, a state v and a formula $\varphi \in \mathrm{ETL}$

$$\mathcal{E}, v \models \varphi?$$

Model-Checking Knowledge and Time!

$\mathcal{E}, v \models \varphi$ is defined as $\mathcal{R}(\mathcal{E}) \models \varphi$
where $\mathcal{R}(\mathcal{E}) = (R, \mathrm{Prop}, \zeta, (\sim_i)_{i \in \underline{n}})$ is the unravelling of $\mathcal{E}$ from v

- R is the set of all runs of $\mathcal{E}$ from v
- $\zeta(\pi, n) = \{P \in \mathrm{Prop} \mid v_n \in P\}$
- $\sim_i$ is either $\sim_i^*$ or $\overleftarrow{\sim}_i^*$

Model Checking Knowledge

Model Checking Knowledge

- Given a finite Kripke structure $\mathcal{K}$, a state v and an epistemic formula φ, does $\mathcal{K}, v \models \varphi$ hold?

Model Checking Knowledge

- Given a finite Kripke structure $\mathcal{K}$, a state v and an epistemic formula φ, does $\mathcal{K}, v \models \varphi$ hold?
  Time $O(||\mathcal{K}|| \cdot |\varphi|)$

Model Checking Knowledge

- Given a finite Kripke structure $\mathcal{K}$, a state v and an epistemic formula φ, does $\mathcal{K}, v \models \varphi$ hold?
  Time $O(||\mathcal{K}|| \cdot |\varphi|)$ (Labeling Algorithm)

Model Checking Knowledge

- Given a finite Kripke structure $\mathcal{K}$, a state v and an epistemic formula φ, does $\mathcal{K}, v \models \varphi$ hold?
  Time $O(||\mathcal{K}|| \cdot |\varphi|)$ (Labeling Algorithm)
- For epistemic formulas with common knowledge still polynomial time

Model Checking Time

Model Checking Time

Given a finite system $\mathcal{E}$, a state v and an LTL formula φ, does $\mathcal{E}, v \models \varphi$ hold?

Model Checking Time

Given a finite system $\mathcal{E}$, a state v and an LTL formula φ, does $\mathcal{E}, v \models \varphi$ hold?
Polynomial Space

Model Checking Time

Given a finite system $\mathcal{E}$, a state v and an LTL formula φ, does $\mathcal{E}, v \models \varphi$ hold?
Polynomial Space
(Translation of LTL-formulas into Büchi automata)

Model Checking Knowledge and Time

For synchronous systems decidable, but non-elementary complexity (van der Meyden, Shilov '99)

Model Checking Knowledge and Time

For synchronous systems decidable, but non-elementary complexity (van der Meyden, Shilov '99)

(1) Reduction to chain logic with equal level predicate

Model Checking Knowledge and Time

For synchronous systems decidable, but non-elementary complexity (van der Meyden, Shilov '99)

(1) Reduction to chain logic with equal level predicate

Model Checking Knowledge and Time

For synchronous systems decidable, but non-elementary complexity (van der Meyden, Shilov '99)

(1) Reduction to chain logic with equal level predicate

(2)
- Use k-trees to interpret formulas of knowledge-depth k
- For fixed k, infinite sequences of k-trees can be recognized by automata
- Also involves a factorization of formulas into temporal and knowledge components
- $\rightsquigarrow$ space polynomial in $|\varphi| \cdot \text{exp}(\text{depth}(\varphi), O(|\mathcal{E}|))$

Model Checking Knowledge and Time

- For synchronous systems and formulas with common knowledge undecidable (van der Meyden, Shilov, '99)
  Until and Common Knowledge allow an arbitrary reach through two orthogonal dimensions of the semantic structures

Model Checking Knowledge and Time

- For synchronous systems and formulas with common knowledge undecidable (van der Meyden, Shilov, '99)
  Until and Common Knowledge allow an arbitrary reach through two orthogonal dimensions of the semantic structures
- For synchronous systems and formulas with common knowledge but *without until* PSPACE-complete (van der Meyden, Shilov, '99)
  Without until, the temporal operators can only look $|\varphi|$ steps into the system

From Verification to Control

From Verification to Control

- We are not only interested in model checking closed systems but in *synthesizing reactive* systems
- Reactive Systems interact with an environment
- The desired behavior of the system is given by a formal specification, for example a temporal formula $\varphi \in \mathsf{LTL}$

From Verification to Control

- We are not only interested in model checking closed systems but in *synthesizing reactive* systems
- Reactive Systems interact with an environment
- The desired behavior of the system is given by a formal specification, for example a temporal formula $\varphi \in$ LTL
- Question: Can we ensure a faultless interaction of the components of the system, independently of the behavior of the environment
- Natural Model: Games on Graphs

Games with Partial Observation

Multi-player games on graphs:

Games with Partial Observation

Multi-player games on graphs:

- A multi-player game is basically a multi-agent system, where each position is controlled either by one of the cooperating players or by the environment
- We view the environment as an additional player
- cooperating player represent components of the system
- Question: Does this coalition have a winning strategy for the game?

Games with Partial Observation

Multi-player games on graphs:

- A multi-player game is basically a multi-agent system, where each position is controlled either by one of the cooperating players or by the environment
- We view the environment as an additional player
- cooperating player represent components of the system
- Question: Does this coalition have a winning strategy for the game?
- The winning condition for the coalition is defined by the specification φ
- We often consider systems with partial observation
- It is then very desirable to be able to refer to the knowlegde of the components in the specification: ETL
- The environment may also have partial information about the system

Games with Partial Observation

A *game with* n *players* has the form

$$\mathcal{G} = (V, (V_i)_{i \in \underline{n}}, \mathrm{Prop}, \Delta, (\sim_i^V)_{i \in \underline{n}}, (\sim_i^A)_{i \in \underline{n}})$$

where

Games with Partial Observation

A *game with* n *players* has the form

$$\mathcal{G} = (V, (V_i)_{i \in \underline{n}}, \mathrm{Prop}, \Delta, (\sim_i^V)_{i \in \underline{n}}, (\sim_i^A)_{i \in \underline{n}})$$

where

- V is the finite set of positions
- V_i is the set of positions of player i
- $\mathrm{Prop} = \{P_v \mid v \in V\}$
- $\Delta \subseteq V \times A \times V$ is the move relation
- $\sim_i^V$ and $\sim_i^A$ are equivalence relations on V and A respectively

Games with Partial Observation

A *game with n players* has the form

$$\mathcal{G} = (V, (V_i)_{i \in \underline{n}}, \mathrm{Prop}, \Delta, (\sim_i^V)_{i \in \underline{n}}, (\sim_i^A)_{i \in \underline{n}})$$

where

- V is the finite set of positions
- V_i is the set of positions of player i
- $\mathrm{Prop} = \{P_v \mid v \in V\}$
- $\Delta \subseteq V \times A \times V$ is the move relation
- $\sim_i^V$ and $\sim_i^A$ are equivalence relations on V and A respectively

1. $v \sim_i w \Rightarrow v, w \in V_i$ or $v, w \notin V_i$
2. $v \sim_i w \Rightarrow \mathrm{act}(v) = \mathrm{act}(w)$
3. $a, b \in A_i$ and $a \neq b \Rightarrow a \not\sim_i^A b$

Games with Partial Observation

- *Play:* Run

Games with Partial Observation

- *Play:* Run
- *Strategy for player i:* function $\sigma_i : (VA)^*V_i \to A$ with

$$\sigma_i(\pi) = \sigma_i(\rho) \text{ for all } \pi, \rho \in (VA)^*V_i \text{ with } \pi \sim_i \rho$$

Games with Partial Observation

- *Play:* Run
- *Strategy for player i:* function $\sigma_i : (VA)^*V_i \to A$ with

$$\sigma_i(\pi) = \sigma_i(\rho) \text{ for all } \pi, \rho \in (VA)^*V_i \text{ with } \pi \sim_i \rho$$

- $\sim_i \in \{\sim_i^*, \overleftarrow{\sim}_i^*\}$

Games with Partial Observation

- *Play:* Run
- *Strategy for player i:* function $\sigma_i : (VA)^*V_i \to A$ with

$$\sigma_i(\pi) = \sigma_i(\rho) \text{ for all } \pi, \rho \in (VA)^*V_i \text{ with } \pi \sim_i \rho$$

- $\sim_i \in \{\sim_i^*, \overleftarrow{\sim}_i^*\}$
- *Joint strategy:* $\sigma = (\sigma_1, \ldots, \sigma_n)$

Games with Partial Observation

- *Play:* Run
- *Strategy for player i:* function $\sigma_i : (VA)^*V_i \to A$ with

$$\sigma_i(\pi) = \sigma_i(\rho) \text{ for all } \pi, \rho \in (VA)^*V_i \text{ with } \pi \sim_i \rho$$

- $\sim_i \in \{\sim_i^*, \overleftarrow{\sim}_i^*\}$
- *Joint strategy:* $\sigma = (\sigma_1, \ldots, \sigma_n)$
- *Winning condition for the coalition:* set $W \subseteq V^\omega$

Games with Partial Observation

- *Play:* Run
- *Strategy for player i:* function $\sigma_i : (VA)^* V_i \to A$ with

$$\sigma_i(\pi) = \sigma_i(\rho) \text{ for all } \pi, \rho \in (VA)^* V_i \text{ with } \pi \sim_i \rho$$

- $\sim_i \in \{\sim_i^*, \overleftarrow{\sim}_i^*\}$
- *Joint strategy:* $\sigma = (\sigma_1, \ldots, \sigma_n)$
- *Winning condition for the coalition:* set $W \subseteq V^\omega$
- *Winning strategy for the coalition:* Joint strategy σ such that each play π which is consistent with σ is won by the coalition

Synthesis from ETL-specifications

Synthesis from ETL-specifications

An ETL-formula φ defines a winning condition

$$L(\varphi) = \{\pi \in (VA)^\omega \mid (\pi, 0) \models \varphi\}$$

Synthesis from ETL-specifications: Two Players

Synthesis from ETL-specifications: Two Players

Given a two-player game $\mathcal{G}$ and an ETL-formula φ, is there a strategy σ_0 for player 0 such that for all plays π which are consistent with σ_0 we have $\pi \in L(\varphi)$?

Synthesis from ETL-specifications: Two Players

Given a two-player game $\mathcal{G}$ and an ETL-formula φ, is there a strategy σ_0 for player 0 such that for all plays π which are consistent with σ_0 we have $\pi \in L(\varphi)$?

- Clearly we should assume player 0 to know his own strategy σ_0
- So the evaluation of the knowledge operator K_0 should be relative to histories (points) which are consistent with σ_0

Synthesis from ETL-specifications: Two Players

Given a two-player game $\mathcal{G}$ and an ETL-formula φ, is there a strategy σ_0 for player 0 such that for all plays π which are consistent with σ_0 we have $\pi \in L(\varphi)$?

- Clearly we should assume player 0 to know his own strategy σ_0
- So the evaluation of the knowledge operator K_0 should be relative to histories (points) which are consistent with σ_0
- However: if $(\pi, n) \sim_0 (\rho, m)$ for $\sim_0 \in \{\sim_0^*, \overleftarrow{\sim}_0^*\}$, then (π, n) is consistent with σ_0 if, and only if, (ρ, m) is consistent with σ_0
- Due to the fact that player 0 can distinguish any two of his own actions

Synthesis from ETL-specifications: Two Players

- For the knowledge operator K_1 to make sense, we have to assume that player 1 does not know the strategy of player 0
- Otherwise player 1 has full information about the history

Synthesis from ETL-specifications: Two Players

- For the knowledge operator K_1 to make sense, we have to assume that player 1 does not know the strategy of player 0
- Otherwise player 1 has full information about the history
- If player 0 models a controller of an environment which is not actually antagonistic but merely unpredictable, then given any strategy σ_0 for player 0, the joint system is constrained by σ_0, so the knowledge of both players should be relative to σ_0

Synthesis from ETL-specifications: Two Players

- For the knowledge operator K_1 to make sense, we have to assume that player 1 does not know the strategy of player 0
- Otherwise player 1 has full information about the history
- If player 0 models a controller of an environment which is not actually antagonistic but merely unpredictable, then given any strategy σ_0 for player 0, the joint system is constrained by σ_0, so the knowledge of both players should be relative to σ_0
- If player 0 models a network server which might interact with a user, then the protocol of the server may be off-limits to the user, so player 1 does not know σ_0
- One requirement for the protocol might be that the user is never able to learn the value of some internal variables of the server
- This can be expressed using K_1

Synthesis from ETL-specifications: Two Players

- Consider a parity game $\mathcal{G}$ with coloring $\mathrm{col} : V \to \{1, \ldots, r\}$

Synthesis from ETL-specifications: Two Players

- Consider a parity game $\mathcal{G}$ with coloring $\mathrm{col} : V \to \{1, \ldots, r\}$
- parity $:= \bigvee_{c \text{ even}} GF\,\mathrm{col}^{-1}(c) \wedge FG \bigwedge_{c' < c} \neg\,\mathrm{col}^{-1}(c')$ defines the parity objective for player 0

Synthesis from ETL-specifications: Two Players

- Consider a parity game $\mathcal{G}$ with coloring $\mathrm{col} : V \to \{1, \ldots, r\}$
- parity $:= \bigvee_{c \text{ even}} GF\,\mathrm{col}^{-1}(c) \wedge FG \bigwedge_{c'<c} \neg\,\mathrm{col}^{-1}(c')$ defines the parity objective for player 0
- $K_0^{\mathrm{col}} := \bigvee_{c \in C} K_0\,\mathrm{col}^{-1}(c)$ says that player 0 knows the color of the recent position

Synthesis from ETL-specifications: Two Players

- Consider a parity game $\mathcal{G}$ with coloring $\mathrm{col} : V \to \{1, \ldots, r\}$
- parity $:= \bigvee_{c \text{ even}} GF\,\mathrm{col}^{-1}(c) \wedge FG \bigwedge_{c' < c} \neg\,\mathrm{col}^{-1}(c')$ defines the parity objective for player 0
- $K_0^{\mathrm{col}} := \bigvee_{c \in C} K_0\,\mathrm{col}^{-1}(c)$ says that player 0 knows the color of the recent position
- So $\varphi =$ parity $\wedge\, G(\neg K_1 K_0^{\mathrm{col}} \wedge \neg K_1 \neg K_0^{\mathrm{col}})$ additionally requires that player 1 never knows whether player 0 knows the recent color

Synthesis from ETL-specifications: Two Players

Synthesis from ETL-specifications: Two Players

- For synchronous finite two-player games with ETL winning conditions which use only the knowledge operator K_0 2EXPTIME-complete (van der Meyden, Vardi '98)

Synthesis from ETL-specifications: Two Players

- For synchronous finite two-player games with ETL winning conditions which use only the knowledge operator K_0 2EXPTIME-complete (van der Meyden, Vardi '98)
- Uses tree automata: A tree automaton can process the tree-representation of a strategy, viewed as a function $\sigma_0 : \mathrm{Obs}_i^* \to A$
- Check that all plays are won by player 0: Universal part
- Evaluate K_0: Knowledge sets

Synthesis from ETL-specifications: Two Players

Synthesis from ETL-specifications: Two Players

Knowledge set of player i at point (π, n):

$$\mathcal{K}_i(\pi, n) = \{\text{last}(\rho, m) \,|\, (\rho, m) \sim_i (\pi, n)\} \subseteq V$$

(Set of positions that player i considers possible at point (π, n))

Synthesis from ETL-specifications: Two Players

Knowledge set of player i at point (π, n):

$$\mathcal{K}_i(\pi, n) = \{\mathrm{last}(\rho, m) \mid (\rho, m) \sim_i (\pi, n)\} \subseteq V$$

(Set of positions that player i considers possible at point (π, n))
Can be computed iteratively:

$$\mathcal{K}_i(\pi, n+1) = \mathrm{Post}_{[a_{n+1}]}(\mathcal{K}_i(\pi, n)) \cap [v_{n+1}]$$

$\rightsquigarrow$ **Powerset Construction** (Reif '84)

Transforms a two-player game with partial information into a two-player game with full information

Synthesis from ETL-specifications: Two Players

Synthesis from ETL-specifications: Two Players

Powerset Construction can be generalized to arbitrary ω-regular winning conditions both in the synchronous and in the asynchronous case ('10)

Synthesis from ETL-specifications: Two Players

Powerset Construction can be generalized to arbitrary ω-regular winning conditions both in the synchronous and in the asynchronous case ('10)

Theorem

Any ETL definable winning condition is ω-regular. ('10)

More precisely: For any game $\mathcal{G}$ and any ETL-formula φ with knowledge-operators K_0 and K_1 which have either $\sim_i^*$ or $\overleftarrow{\sim}_i^*$ semantics, we can effectively construct an S1S-formula $\psi(x)$ such that for any play π and any $n \in \mathbb{N}$ we have

$$(\pi, n) \models \varphi \iff \pi \models \psi(n)$$

S1S: Monadic Second Order Logic interpreted in word structures $(\mathbb{N}, (P_a)_{a \in \Sigma}, <)$

Synthesis from ETL-specifications: Two Players

Synthesis from ETL-specifications: Two Players

- Highlights the view of $L(\varphi)$ as a winning condition (set of plays)

Synthesis from ETL-specifications: Two Players

- Highlights the view of $L(\varphi)$ as a winning condition (set of plays)
- Makes the powerset construction applicable to ETL winning conditions

Synthesis from ETL-specifications: Two Players

- Highlights the view of $L(\varphi)$ as a winning condition (set of plays)
- Makes the powerset construction applicable to ETL winning conditions
- Also holds for the asynchronous case

Synthesis from ETL-specifications: Two Players

Proof.

Synthesis from ETL-specifications: Two Players

Proof.

- By induction on the structure of φ
- Interesting case: $K\varphi$

Synthesis from ETL-specifications: Two Players

Proof.

- By induction on the structure of φ
- Interesting case: $K\varphi$
- $(\pi, n) \models \neg K\varphi$ iff there is some $(\rho, m) \sim (\pi, n)$: $(\rho, m) \models \neg\varphi$

Synthesis from ETL-specifications: Two Players

Proof.

- By induction on the structure of φ
- Interesting case: $K\varphi$
- $(\pi, n) \models \neg K\varphi$ iff there is some $(\rho, m) \sim (\pi, n)$: $(\rho, m) \models \neg\varphi$

$$
\begin{aligned}
\neg\exists X_{va} \quad [\quad & \forall y(\bigvee_{va}(X_{va}y \wedge \bigwedge_{wb\neq va} \neg X_{wb}y)) \\
\wedge \quad & \forall y \forall z(Sy = z \rightarrow \bigvee_{(v,w)\in E_a,\, b\in A} X_{va}y \wedge X_{wb}z) \\
\wedge \quad & \forall (y \leq x)(\bigwedge_{va}(P_{va}y \rightarrow \bigvee_{wb\sim va} X_{wb}y)) \\
\wedge \quad & \neg\psi(P_{va}/X_{va}) \quad]
\end{aligned}
$$

Synthesis from ETL-specifications: Two Players

Synthesis from ETL-specifications: Two Players

- In the asynchronous case?

Synthesis from ETL-specifications: Two Players

- In the asynchronous case?
- Use automata!

Synthesis from ETL-specifications: Two Players

- In the asynchronous case?
- Use automata!
- A nondeterministic ω-automaton can guess such a (ρ, m) asynchronously
- In this particular asynchronous case the automaton can be constructed effectively

Synthesis from ETL-specifications: Two Players

Corollary

The asynchronous synthesis problem for ETL specifications with knowledge operators K_0 and K_1 is decidable.

Synthesis from ETL-specifications

Given a game $\mathcal{G}$ and an ETL-formula φ, is there a joint strategy σ for the coalition such that for all plays π which are consistent with σ we have $\pi \in L(\varphi)$?

Synthesis from ETL-specifications

Given a game $\mathcal{G}$ and an ETL-formula φ, is there a joint strategy σ for the coalition such that for all plays π which are consistent with σ we have $\pi \in L(\varphi)$?

- For synchronous 3-player games with LTL winning conditions undecidable (Pnueli, Rosner '90, Reif '01)

Synthesis from ETL-specifications

Given a game $\mathcal{G}$ and an ETL-formula φ, is there a joint strategy σ for the coalition such that for all plays π which are consistent with σ we have $\pi \in L(\varphi)$?

- For synchronous 3-player games with LTL winning conditions undecidable (Pnueli, Rosner '90, Reif '01)
- For synchronous hierarchical n-player games with LTL winning conditions decidable (Pnueli, Rosner '90, van der Meyden, Wilke '05)

Synthesis from ETL-specifications

Given a game $\mathcal{G}$ and an ETL-formula φ, is there a joint strategy σ for the coalition such that for all plays π which are consistent with σ we have $\pi \in L(\varphi)$?

- For synchronous 3-player games with LTL winning conditions undecidable (Pnueli, Rosner '90, Reif '01)
- For synchronous hierarchical n-player games with LTL winning conditions decidable (Pnueli, Rosner '90, van der Meyden, Wilke '05)
- For synchronous 3-player games with ETL winning conditions where only player 0 has partial observation undecidable (van der Meyden, Wilke '05)

Synthesis from ETL-specifications

On the other hand:

- Omega-Regularity of ETL winning conditions also holds for n agents (same proof)

Synthesis from ETL-specifications

On the other hand:

- Omega-Regularity of ETL winning conditions also holds for n agents (same proof)
- Omega-Regular n player games with can be transformed into parity games

Synthesis from ETL-specifications

On the other hand:

- Omega-Regularity of ETL winning conditions also holds for n agents (same proof)
- Omega-Regular n player games with can be transformed into parity games
- such that the hierarchy of knowledge if preserved

Synthesis from ETL-specifications

On the other hand:

- Omega-Regularity of ETL winning conditions also holds for n agents (same proof)
- Omega-Regular n player games with can be transformed into parity games
- such that the hierarchy of knowledge if preserved
- Parity Objectives are LTL-definable

Synthesis from ETL-specifications

On the other hand:

- Omega-Regularity of ETL winning conditions also holds for n agents (same proof)
- Omega-Regular n player games with can be transformed into parity games
- such that the hierarchy of knowledge if preserved
- Parity Objectives are LTL-definable
- Hierarchical games with ETL winning conditions are decidable

Synthesis from ETL-specifications

Reason:

Synthesis from ETL-specifications

Reason:
The actual question van der Meyden, Vardi and Wilke ask is

> Given a game $\mathcal{G}$ and an ETL specification φ, is there a joint strategy σ for the coalition such that $\mathcal{R}(\mathcal{G}, \sigma) \models \varphi$?

where $\mathcal{R}(\mathcal{G}, \sigma)$ is the unravelling of $\mathcal{G}$ with respect to σ!

Synthesis from ETL-specifications

Reason:
The actual question van der Meyden, Vardi and Wilke ask is

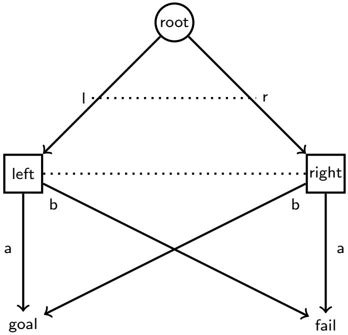
> Given a game $\mathcal{G}$ and an ETL specification φ, is there a joint strategy σ for the coalition such that $\mathcal{R}(\mathcal{G}, \sigma) \models \varphi$?

where $\mathcal{R}(\mathcal{G}, \sigma)$ is the unravelling of $\mathcal{G}$ with respect to σ!

- There, the evaluation of knowledge operators is relative to plays which are consistent with σ
- So all players know the strategy of any other cooperating player
- In the two-player case this is equivalent but in the multiplayer case it makes a difference

Cooperation in Multi-Player Games

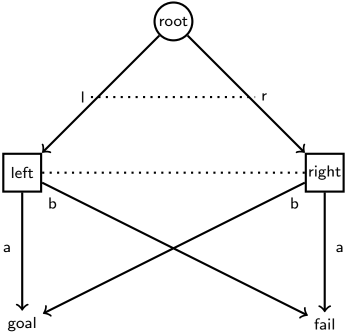
Knowing the strategies of your companions is different from *relying* on their strategies:



Cooperation in Multi-Player Games

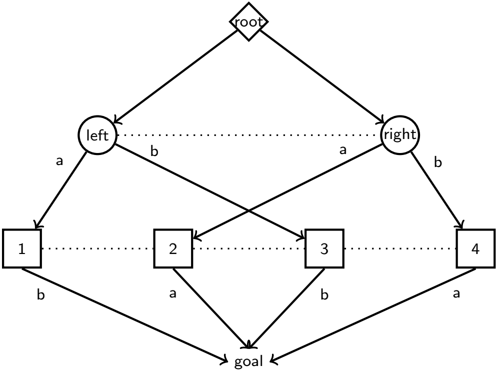
Knowing the strategies of your companions is different from *relying* on their strategies:

$\mathcal{K}_1 = \{\text{left}, \text{right}\} : \neg K_1 P_{\text{left}}$



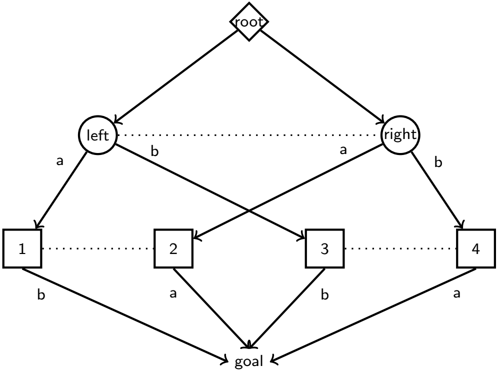
Cooperation in Multi-Player Games

Observing the actions of your companions is not enough for *knowing* their strategies:



Cooperation in Multi-Player Games

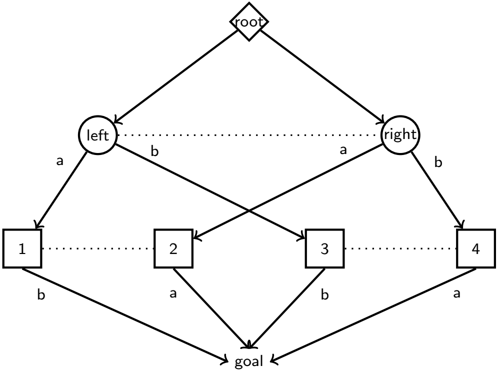
Observing the actions of your companions is not enough for *knowing* their strategies:



Cooperation in Multi-Player Games

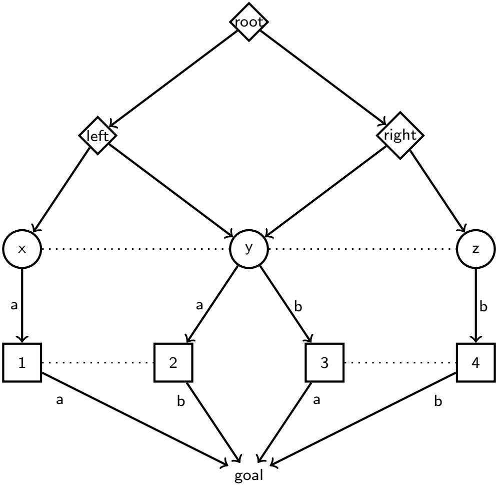
Observing the actions of your companions is not enough for *knowing* their strategies:

$\mathcal{K} = \{1, 2\} : \neg K_1 P_1$



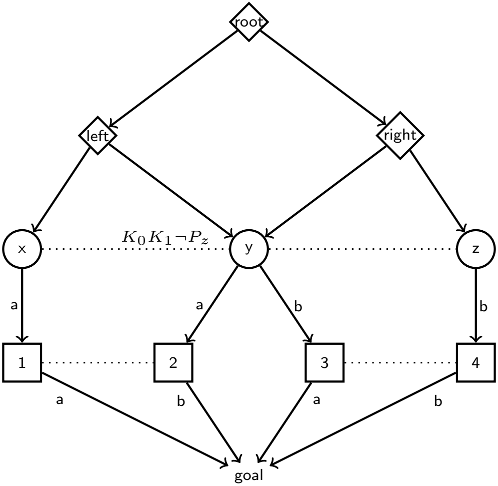
Cooperation in Multi-Player Games

Your knowledge-set does not provide you sufficient information, even if you *do know* the strategies of your companions:



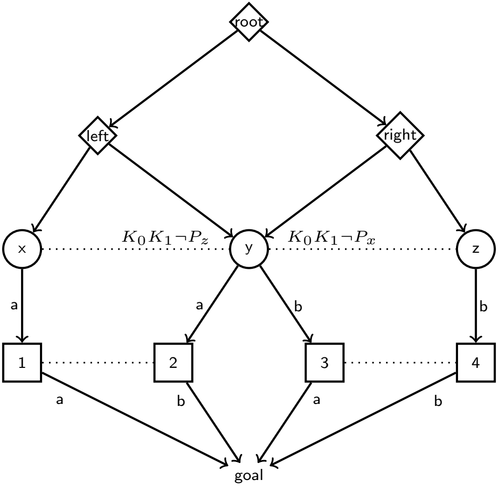
Cooperation in Multi-Player Games

Your knowledge-set does not provide you sufficient information, even if you *do know* the strategies of your companions:



Cooperation in Multi-Player Games

Your knowledge-set does not provide you sufficient information, even if you *do know* the strategies of your companions:



Cooperation in Multi-Player Games

Your knowledge-set does not provide you sufficient information, even if you *do know* the strategies of your companions: