Modal Logics of Strategic Interaction

Workshop LSB@NASSLLI 2016, Rutgers University

Delft University of Technology

Helle Hvid Hansen

14 July 2016

Outline

Outline

TU Delft

Logic for Social Software

Social Software — term coined by Parikh.

Logic in Computer Science	Logic for Social Software
Specification and verification of computer systems	Specification and verification of social procedures

Logic for Social Software

Social Software — term coined by Parikh.

Logic in Computer Science	Logic for Social Software
Specification and verification of computer systems	Specification and verification of social procedures

Algorithms, complexity:

- Algorithmic Game Theory
- Computational Social Choice

Logic for Social Software

Social Software — term coined by Parikh.

Logic in Computer Science	Logic for Social Software
Specification and verification of computer systems	Specification and verification of social procedures

Algorithms, complexity:

- Algorithmic Game Theory
- Computational Social Choice

Logic approach: Translate design problems into logic questions: e.g. mechanism design via satisfiability.

Modal Logics of Strategic Interaction

- Coalition Logic (Pauly)
- Game Logic (Parikh)
- Alternating-time Temporal logic (Alur, Henzinger, Kupferman)
- Strategy Logic (Chatterjee et al.)
- ... (modal logics, AI literature)

Conferences: LOFT, TARK, WoLLiC, LORI, AAMAS, IJCAI, ECAI, JELIA, ...

Outline

Angelina, Edwin and the Judge

Example: Formalising constitutions (cf. Allan Gibbard, 1974)

Angelina has to decide whether she wants to marry Edwin, the (male) judge, or stay single. Edwin and the judge each can similarly decide whether they want to stay single or marry Angelina.

Constitution:
Everyone has the right to choose who they marry and to choose to stay single.

Angelina, Edwin and the Judge

Specify the rights (or abilities) of players as sets of outcomes that they can "force".

Players $N = \{a, e, j\}$
States $S = \{m_0, m_e, m_j\}$ representing

- m_0: Angelina stays single,
- m_e: Angelina marries Edwin,
- m_j: Angelina marries the judge.

Players can force the outcome in some subset of states:

player(s)	can force
a	$\{m_0\}, \{m_0, m_e\}, \{m_0, m_j\}$
e	$\{m_0, m_j\}$
j	$\{m_0, m_e\}$

player(s)	can force
$\{a, e\}$	$\{m_e\}$
$\{a, j\}$	$\{m_j\}$
$\{e, j\}$	$\{m_0\}$

Game Theoretic Modelling

Angelina chooses the table.
Edwin chooses the row.
Judge chooses the column.

	s	m
s	m_0	m_0
m	m_0	m_0

Stay single

	s	m
s	m_0	m_0
m	m_e	m_e

Marry Edwin

	s	m
s	m_0	m_j
m	m_0	m_j

Marry Judge

Note: No preferences over outcomes are specified.

Effectivity in Strategic Games

Def. A *game form* $G = (N, \{\Sigma_i \mid i \in N\}, S, o\colon \Sigma \to S)$ consists of

- a non-empty, finite set N of players,
- for each $i \in N$, a non-empty (finite) set of strategies Σ_i,
- a set S of states,
- an outcome function $o\colon \prod_i \Sigma_i \to S$

TU Delft

Effectivity in Strategic Games

Def. A *game form* $G = (N, \{\Sigma_i \mid i \in N\}, S, o\colon \Sigma \to S)$ consists of

- a non-empty, finite set N of players,
- for each $i \in N$, a non-empty (finite) set of strategies Σ_i,
- a set S of states,
- an outcome function $o\colon \prod_i \Sigma_i \to S$

Def. The effectivity function of a game form G is the function $E_G\colon \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ defined by:

$$X \in E_G(C) \quad \text{iff} \quad \exists \sigma_C \forall \sigma_{\overline{C}} : o(\sigma_C, \sigma_{\overline{C}}) \in X.$$

(α-effectivity, as opposed to β-effecivity $\forall \ldots \exists \ldots$)

$E_G(C)$ is the collection of subsets for which C is effective.

TU Delft

Properties of Effectivity Functions

For any game form G, the effectivity function $E_G : \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ is

1. outcome-monotonic: for all $C \subseteq N$,
   if $X_1 \subseteq X_2 \subseteq S$ and $X_1 \in E(C)$ then $X_2 \in E(C)$.
2. coalition-monotonic: if $C_1 \subseteq C_2$ then $E(C_1) \subseteq E(C_2)$.
3. regular: for all $C \subseteq N$, if $X \in E(C)$ then $\overline{X} \notin E(\overline{C})$.
4. maximal: for all $C \subseteq N$, if $\overline{X} \notin E(\overline{C})$ then $X \in E(C)$.
5. superadditive: for all $C_1, C_2 \subseteq N$ such that $C_1 \cap C_2 = \emptyset$,
   if $X_1 \in E(C_1)$ and $X_2 \in E(C_2)$ then $X_1 \cap X_2 \in E(C_1 \cup C_2)$.

TUDelft

Playable Effectivity Functions

When is a function $E\colon \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ the effectivity function of some game form G?

Playable Effectivity Functions

When is a function $E\colon \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ the effectivity function of some game form G?

Def. A function $E\colon \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ is *playable* if

1. for all $C \subseteq N$: $\emptyset \notin E(C)$
2. for all $C \subseteq N$: $S \in E(C)$ (non-empty)
3. E is N-maximal: for all $X \subseteq S$, if $\overline{X} \notin E(\emptyset)$ then $X \in E(N)$
4. E is superadditive.

Theorem (Pauly): E is playable iff there is a G such that $E = E_G$.
(In particular, playability implies monotonicity and regularity.)

Dynamic Models of Coalitional Effectivity

- Extensive game with simultaneous moves is $\gamma\colon S \to \Gamma(N, S)$ where $\Gamma(N, S)$ are all game forms for N and S.
  At each state, a strategic game is played that determines the next state.

Dynamic Models of Coalitional Effectivity

- Extensive game with simultaneous moves is $\gamma\colon S \to \Gamma(N, S)$ where $\Gamma(N, S)$ are all game forms for N and S.
  At each state, a strategic game is played that determines the next state.
- Def. A *(coalitional) effectivity frame* $F = (N, S, E)$ consists of a set S of states and a function

$$E\colon S \to \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$$

such that for all $s \in S$, $E(s)\colon \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ is playable.

Notation: $sE_C X$ means $X \in E(s)(C)$
("at state s, coalition C is effective for X").

Coalition Logic: Syntax & Semantics

- Def. Let Φ_0 be a set of atomic propositions.
  An effectivity model $M = (N, S, E, V)$ is an effectivity frame (N, S, E) together with a valuation $V\colon \Phi_0 \to \mathcal{P}(S)$.
- Coalition Logic syntax $\mathcal{L}(N, \Phi_0)$:

$$\varphi \quad ::= \quad p \in \Phi_0 \mid \neg\varphi \mid \varphi \wedge \varphi \mid [C]\varphi, \quad C \subseteq N$$

TU Delft

Coalition Logic: Syntax & Semantics

- Def. Let Φ_0 be a set of atomic propositions.
  An effectivity model $M = (N, S, E, V)$ is an effectivity frame (N, S, E) together with a valuation $V\colon \Phi_0 \to \mathcal{P}(S)$.
- Coalition Logic syntax $\mathcal{L}(N, \Phi_0)$:

$$\varphi \quad ::= \quad p \in \Phi_0 \mid \neg\varphi \mid \varphi \wedge \varphi \mid [C]\varphi, \quad C \subseteq N$$

- Coalition Logic semantics: Let $M = (N, S, E, V)$ be an effectivity model, and $s \in S$.

$$\begin{array}{lcl} M, s \models p & \iff & s \in V(p) \\ M, s \models \neg\varphi & \iff & M, s \not\models \varphi \\ M, s \models \varphi \wedge \psi & \iff & M, s \models \varphi \text{ and } M, s \models \psi \\ M, s \models [C]\varphi & \iff & [\![\varphi]\!]_M \in E_C(s) \end{array}$$

where $[\![\varphi]\!]_M = \{t \in S \mid M, t \models \varphi\}$.

TU Delft

Coalition Logic: Axiomatisation

Def. Let $\mathbf{CL}_N$ denote the smallest set of $\mathcal{L}(N, \Phi_0)$-formulas that

- contains all propositional tautologies,
- contains the following axiom (schemas):

$$
\begin{array}{ll}
(\bot) & \neg[C]\bot \\
(\top) & [C]\top \\
(\mathtt{N}) & \neg[\emptyset]\neg\varphi \to [N]\varphi \\
(\mathtt{M}) & [C](\varphi \land \psi) \to [C]\varphi \\
(\mathtt{S}) & ([C_1]\varphi_1 \land [C_2]\varphi_2) \to [C_1 \cup C_2](\varphi_1 \land \varphi_2) \\
 & \text{ where } C_1 \cap C_2 = \emptyset.
\end{array}
$$

- is closed under modus ponens and the congruence rule for all $C \subseteq N$:

$$
\frac{\varphi \quad \varphi \to \psi}{\psi} \qquad\qquad \frac{\varphi \leftrightarrow \psi}{[C]\varphi \leftrightarrow [C]\psi}
$$

Completeness and Decidability

Theorem (Pauly): $\mathbf{CL}_N$ is sound and complete w.r.t. the class of all coalitional effectivity models.

Theorem (Pauly): The satisfiability problem (i.e. given formula φ, is there a model in which φ is true?) is decidable in PSPACE.

Method: Use satisfiability to find implementation of a constitution specified by φ.

Extended of Coalition Logic

(cf. Pauly, 2001):

Add fixpoint (temporal) operators:

$[C^*]\varphi$ "C can eventually bring about φ".
$[C^\times]\varphi$ "C can maintain φ in the long run".

Coalition Logic and Extended Coalition Logic can be translated into Alternating-time Temporal Logic (Goranko).

Nash-consistent Coalition Logic

(cf. Pauly & Hansen, 2004):

Add to $\mathbf{CL}_N$ the rule:

$$\frac{\bigvee_{i \in N} \varphi_i}{\bigvee_{i \in N}[N \setminus \{i\}]\varphi_i} NC$$

Nash-consistent Coalition Logic is sound and complete w.r.t. the class of Nash-consistent models, i.e., if all (implicit) game forms in the model have a Nash equilibrium for any set of preferences the players may have over outcome states.

Application: Gibbard's Paradox

- Two players $N = \{a, b\}$ (Ann and Bob)
- Each player can choose which colour shirt to wear: white or blue.
- $\Phi_0 = \{p_a \text{ (Ann wears white)}, p_b \text{ (Bob wears white)}\}$.
- Constitution:

  Everyone has the right to decide on the colour of his/her own shirt:

  $$\varphi^+ = [a]p_a \wedge [a]\neg p_a \wedge [b]p_b \wedge [b]\neg p_b$$

  No single agent is effective for coordination or anti-coordination:

  $$\begin{aligned} \varphi^- = \quad & \textstyle\bigwedge_{i\in\{a,b\}} \neg[i]((p_a \wedge p_b) \vee (\neg p_a \wedge \neg p_b)) \wedge \\ & \textstyle\bigwedge_{i\in\{a,b\}} \neg[i]((p_a \wedge \neg p_b) \vee (\neg p_a \wedge p_b)) \end{aligned}$$

TU Delft

Application: Gibbard's Paradox (II)

Can the constitution $\varphi = \varphi^+ \wedge \varphi^-$ be implemented?
That is, is φ satsifiable?

The situation can be described by the game form G below on the left:

	w	b
w	(w, w)	(w, b)
b	(b, w)	(b, b)

TUDelft

Application: Gibbard's Paradox (II)

Can the constitution $\varphi = \varphi^+ \wedge \varphi^-$ be implemented?
That is, is φ satsifiable?

The situation can be described by the game form G below on the left:

	w	b
w	(w, w)	(w, b)
b	(b, w)	(b, b)

	w	b
w	$(2, 4)$	$(4, 1)$
b	$(3, 2)$	$(1, 3)$

G is not Nash-consistent. Consider scenario where:
Bob is conformist, and besides that he prefers white to blue.
Ann is non-conformist, and besides that she prefers white to blue.
This is modelled by the payoff matrix on the right. No Nash-equilibrium.

Application: Gibbard's Paradox (III)

We can prove that φ is not satisfiable in a Nash-consistent model, since it is inconsistent:
From tautology

$$((p_a \wedge p_b) \vee (\neg p_a \wedge \neg p_b)) \vee ((p_a \wedge \neg p_b) \vee (\neg p_a \wedge p_b))$$

infer using NC rule

$$[b]((p_a \wedge p_b) \vee (\neg p_a \wedge \neg p_b)) \vee [a]((p_a \wedge \neg p_b) \vee (\neg p_a \wedge p_b))$$

which contradicts φ^-.

Coalition Logic References

- M. Pauly. *A modal logic for coalitional power in games.* Journal of Logic and Computation, 12(1):149166, 2002.
- M. Pauly. *Logic for Social Software*. PhD Thesis, University of Amsterdam, 2001.
- H.H. Hansen & M. Pauly, *Axiomatising Nash-Consistent Coalition Logic*. JELIA 2002, LNAI 2424, Springer, 2002.

- Several more recent extensions: epistemic, announcements, quantification over coalitions, resource-bounded,... (cf. Ågotnes, Alechina, Van Ditmarsch, Van der Hoek, Wooldridge,...)

Outline

From Effectivity to Neighbourhoods

- For every state s,

$$E(s)\colon \mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$$

  is an effectivity function
- For every coalition C,

$$E_C\colon S \to \mathcal{P}(\mathcal{P}(S))$$

  is a *neighbourhood function*.
- A coalitional effectivity frame (N, S, E) can be viewed as a coalition-indexed neighbourhood frame:

$$(S, \{E_C\colon S \to \mathcal{P}(\mathcal{P}(S)) \mid C \subseteq N\})$$

Neighbourhood Semantics of Modal Logic

- A neighbourhood model $M = (S, \nu, V)$ consists of:
  - state space S,
  - neighbourhood function $\nu : S \to \mathcal{P}(\mathcal{P}(S))$,
  - valuation $V : \Phi_0 \to \mathcal{P}(S)$
- Basic Modal Language $\mathcal{L}(\Phi_0)$:

$$\varphi \quad ::= \quad p \in \Phi_0 \mid \neg\varphi \mid \varphi \wedge \varphi \mid \Box\varphi$$

- Neighbourhood semantics of modal logic:

$$\begin{array}{lcl} M, s \models p & \iff & s \in V(p) \\ M, s \models \neg\varphi & \iff & M, s \not\models \varphi \\ M, s \models \varphi \wedge \psi & \iff & M, s \models \varphi \text{ and } M, s \models \psi \\ M, s \models \Box\varphi & \iff & [\![\varphi]\!]_M \in \nu(s) \end{array}$$

where $[\![\varphi]\!]_M = \{t \in S \mid M, t \models \varphi\}$.

TU Delft

Why can't we use good old Kripke models?

- The principle

$$[C]\varphi \wedge [C]\psi \to [C](\varphi \wedge \psi)$$

  is generally NOT valid, because φ-strategy and ψ-strategy may be incompatible.
- But formula $\Box p \wedge \Box q \leftrightarrow \Box(p \wedge q)$ is valid in all Kripke models (i.e. true at all states in all models).
- Not so in neighbourhood models:

$$[\![p]\!] \in \nu(s), [\![q]\!] \in \nu(s) \not\Rightarrow [\![p \wedge q]\!] = [\![p]\!] \cap [\![q]\!] \in \nu(s)$$

  (since $\nu(s)$ need not be closed under intersection).
- If for all s, $\nu(s)$ is upward closed, i.e., $X \in \nu(s), X \subseteq Y$ implies $Y \in \nu(s)$, then $\Box$ is monotonic:

$\Box(p \wedge q) \to \Box p \wedge \Box q$ is valid, or equiv. $\dfrac{\varphi \to \psi}{\Box\varphi \to \Box\psi}$ is valid

TUDelft

Non-normal Modal Logic

- Logics that are weaker than **K** (modal logic of all Kripke frames) are called non-normal.
- Non-normal modal logics were studied early on (cf. Segerberg, Chellas), mainly from a proof-theoretic perspective.
- Classical modal logic is the logic of all neighbourhood models:
  - All propositional tautologies
  - Closed under modus ponens and *congruence rule*: $\dfrac{\varphi \leftrightarrow \psi}{\Box\varphi \leftrightarrow \Box\psi}$
- Monotonic modal logic is the logic of all monotonic neighbourhood models.
  - All propositional tautologies
  - Closed under modus ponens and *monotonicity rule*: $\dfrac{\varphi \rightarrow \psi}{\Box\varphi \rightarrow \Box\psi}$

TU Delft

Theory of Neighbourhood Structures

- Bounded morphisms ✓
- Bisimulation ✓
- Van Benthem Characterisation Theorem ✓
- Algebraic duality ✓
- ...

(cf. Chellas 1980, Segerberg 1971, Hansen, Kupke, Pacuit, ...)

Simulating Non-normal Modal Logics

We can translate non-normal modal logic into normal multi-modal logic [Kracht & Wolter]. Basic idea:

Simulating Non-normal Modal Logics

We can translate non-normal modal logic into normal multi-modal logic [Kracht & Wolter]. Basic idea:

Monotonic modal logic:

$$
\begin{array}{llll}
\textit{Syntax}: & \Box p & \rightsquigarrow & \langle\nu\rangle[\ni]p \\
\textit{Semantics}: & (S,\nu) & \rightsquigarrow & (S \cup \mathcal{P}(S), R_\nu, R_\ni, P_S) \\
 & & \textit{where} & R_\nu \subseteq S \times \mathcal{P}(S) : (s,U) \in R_\nu \text{ iff } U \in \nu(s) \\
 & & & R_\ni \subseteq \mathcal{P}(S) \times S : (U,s) \in R_\ni \text{ iff } U \ni s
\end{array}
$$

Simulating Non-normal Modal Logics

We can translate non-normal modal logic into normal multi-modal logic [Kracht & Wolter]. Basic idea:

Monotonic modal logic:

$$\begin{array}{llll} \textit{Syntax}: & \Box p & \rightsquigarrow & \langle \nu \rangle [\ni] p \\ \textit{Semantics}: & (S, \nu) & \rightsquigarrow & (S \cup \mathcal{P}(S), R_\nu, R_\ni, P_S) \\ & & \textit{where} & R_\nu \subseteq S \times \mathcal{P}(S) : (s, U) \in R_\nu \text{ iff } U \in \nu(s) \\ & & & R_\ni \subseteq \mathcal{P}(S) \times S : (U, s) \in R_\ni \text{ iff } U \ni s \end{array}$$

Classical modal logic:

$$\begin{array}{llll} \textit{Syntax}: & \Box p & \rightsquigarrow & \langle \nu \rangle ([\ni] p \wedge [\not\ni] \neg p) \\ \textit{Semantics}: & (S, \nu) & \rightsquigarrow & (S \cup \mathcal{P}(S), R_\nu, R_\ni, R_{\not\ni}, P_S) \end{array}$$

(Interesting for correspondence theory, but not for completeness etc.)

Neighbourhood Semantics References

- B. F. Chellas. Modal logic, an intoduction. Cambridge University Press, 1980.
- K. Segerberg. An Esssay in Classical Modal Logic. Number 13 in Filosofiska Studier. Uppsala Universitet, 1971.
- M. Kracht and F. Wolter. Normal monomodal logics can simulate all others. Journal of Symbolic Logic, 64(1):99138, 1999.
- H. H. Hansen. Monotonic Modal Logics. MSc Thesis Mathematics. Universiteit van Amsterdam, 2003. Available as ILLC technical report PP-2003-24.
- Helle Hvid Hansen, Clemens Kupke and Eric Pacuit. Neighbourhood Structures: Bisimilarity and Basic Model Theory. Logical Methods in Computer Science, vol. 5, issue 2, paper 2, 2009.

Outline

An External View: Dynamic Logics

- Logic for Social Software: From program logics to game logics.
- External (compositional) view: games are an explicit part of the syntax.
- Internal (monolithic) view: LTL, CTL.

Reasoning About Programs

Consider the following program which takes as input a pair of integers in the variables x and y,

```
while y ≠ 0 do
    begin
        z := x mod y;
        x := y;
        y := z
    end
```

What is the value of x when the program halts?

TU Delft

Reasoning About Programs

Consider the following program which takes as input a pair of integers in the variables x and y,

```
while y ≠ 0 do
    begin
        z := x mod y;
        x := y;
        y := z
    end
```

What is the value of x when the program halts?
Euclidean algorithm for computing the greatest common divisor (GCD) of x and y.

Example Execution

Initial state (x,y,z) = (15,27,0).

command	$y \neq 0;$	$z := x \text{ mod } y;$	$x := y;$	$y := z;$
state	*true*	$(15, 27, 15)$	$(27, 27, 15)$	$(27, 15, 15)$

Example Execution

Initial state (x,y,z) = (15,27,0).

command	$y \neq 0$;	$z := x \bmod y$;	$x := y$;	$y := z$;
state	*true*	$(15, 27, 15)$	$(27, 27, 15)$	$(27, 15, 15)$
command	$y \neq 0$;	$z := x \bmod y$;	$x := y$;	$y := z$;
state	*true*	$(27, 15, 12)$	$(15, 15, 12)$	$(15, 12, 12)$
command	$y \neq 0$;	$z := x \bmod y$;	$x := y$;	$y := z$;
state	*true*	$(15, 12, 3)$	$(12, 12, 3)$	$(12, 3, 3)$
command	$y \neq 0$;	$z := x \bmod y$;	$x := y$;	$y := z$;
state	*true*	$(12, 3, 0)$	$(3, 3, 0)$	$(3, 0, 0)$
command	$y \neq 0$;			
state	*false*			

Return 3.

Basic (Imperative) Programming Constructs

While-language:

- x := t, where t is a term (simple assignments)
- $\alpha; \beta$ (sequential composition)
- if φ then α else β (conditional)
- while φ do α (loop)

TU Delft

PDL Programs and Formulas

Assume given:

- a set A_0 of atomic programs, and
- a set Φ_0 of atomic propositions

The set of PDL programs and formulas (tests) is defined by mutual induction:

$$\begin{array}{lll} \text{formulas } \Phi \ni \varphi & ::= & p \in \Phi_0 \mid \neg\varphi \mid \varphi \vee \varphi \mid [\alpha]\varphi \\ \text{programs } A \ni \alpha & ::= & a \in A_0 \mid \alpha;\alpha \mid \alpha \cup \alpha \mid \alpha^* \mid \varphi? \end{array}$$

PDL program constructs:

- $\alpha;\beta$ is sequential composition (do α and then β)
- $\alpha \cup \beta$ is nondeterministic choice between α and β.
- α^* is iteration of α (execute α a nondeterministically chosen finite number of times)

PDL Programs and Formulas (II)

- PDL programs subsume the While-language:

$$\text{if } \varphi \text{ then } \alpha \text{ else } \beta \stackrel{def}{=} \varphi?; \alpha \cup \neg\varphi?; \beta$$

$$\text{while } \varphi \text{ do } \alpha \stackrel{def}{=} (\varphi?; \alpha)^*; \neg\varphi?$$

- PDL subsumes Hoare logic (weakest precondition calculus)

$$\{\varphi\}\alpha\{\psi\} \quad \leftrightsquigarrow \quad \varphi \to [\alpha]\psi$$

if φ holds, then after executing α, ψ holds

($[\alpha]\psi$ is the weakest precondition for ψ.)

- PDL subsumes Kleene Algebra with Tests:

$$\alpha \equiv \beta \quad \leftrightsquigarrow \quad [\alpha]p \leftrightarrow [\beta]p$$

TU Delft

PDL Models

PDL was introduced by Fischer & Ladner, 1977.

$[\alpha]\varphi$ "after all successful executions of program α, φ holds"

Nondeterministic programs are modelled as relations.

A PDL model $M = (S, \{R_\alpha \mid \alpha \in A\}, V)$ consists of

- a set S (state space),
- a relation $R_\alpha \subseteq S \times S$ for all $\alpha \in A_0$.
- a valuation $V \colon \Phi_0 \to \mathcal{P}(S)$

PDL Semantics

- Interpretation of formulas (as usual):

$$M, s \models [\alpha]\varphi \quad \text{iff} \quad \text{for all } t \in S \text{ s.t. } (s,t) \in R_\alpha : M, t \models \varphi.$$

- Interpretation of programs via relation algebra:

$$\begin{array}{rcl} R_{\alpha;\beta} & = & R_\alpha \circ R_\beta \ \text{(relation composition)} \\ R_{\alpha\cup\beta} & = & R_\alpha \cup R_\beta \\ R_{\alpha^*} & = & R_\alpha^* \ \text{(reflexive, transitive closure)} \\ R_{\varphi?} & = & \{(s,s) \mid s \in [\![\varphi]\!]\} \end{array}$$

A PDL model is a special kind of multi-modal Kripke model.

PDL Axiomatisation

Def. Let **PDL** be the smallest set of formulas which contains

- normal modal logic $\mathbf{K}([\alpha])$ for all $\alpha \in A$
- the following axiom schemas:

$$[\alpha;\beta]\varphi \leftrightarrow [\alpha][\beta]\varphi \qquad [\alpha \cup \beta]\varphi \leftrightarrow [\alpha]\varphi \wedge [\beta]\varphi$$

$$[\psi?]\varphi \leftrightarrow (\psi \to \varphi)$$

$$\varphi \wedge [\alpha][\alpha^*]\varphi \leftrightarrow [\alpha^*]\varphi \qquad \varphi \wedge [\alpha^*](\varphi \to [\alpha]\varphi) \to [\alpha^*]\varphi$$

- and is closed under modus ponens.

Theorem (Kozen & Parikh, 1981): **PDL** is sound and complete with respect to the class of PDL models.

Game Logic (GL)

- Parikh, 1985. Strategic ability in determined 2-player games.

$\langle\gamma\rangle\varphi$ "player 1 has strategy in γ to ensure outcome satisfies φ"

Game Logic (GL)

- Parikh, 1985. Strategic ability in determined 2-player games.

  $\langle\gamma\rangle\varphi$ "player 1 has strategy in γ to ensure outcome satisfies φ"
- Determined 2-player game γ is modelled as a monotone neighbourhood function $E_\gamma \colon S \to \mathcal{P}(\mathcal{P}(S))$ where

  $U \in E_\gamma(s)$ means "player 1 is effective for U in γ at state s".
- Determinacy means that

  $\overline{U} \notin E_\gamma(s)$ iff "player 2 is effective for U in game γ at state s".

Models for Game Logic

Assume given:

- a set Γ_0 of atomic games, and
- a set Φ_0 of atomic propositions

Def. A GL model $M = (S, \{E_\gamma \mid \gamma \in \Gamma_0\}, V)$ consists of

- a set S of states,
- for each atomic game $\gamma \in \Gamma_0$, $E_\gamma : S \to \mathcal{P}\mathcal{P}(S)$ a monotonic neighbourhood function,

$$\text{if } U \in E_\gamma(s) \text{ and } U \subseteq U' \text{ then } U' \in E_\gamma(s).$$

- a valuation $V : \Phi_0 \to \mathcal{P}(S)$.

TU Delft

Syntactic Game Constructs

PDL constructs extended with dual operation on games:

- $\gamma_1;\gamma_2$: play γ_1 then γ_2,
- $\gamma_1 \cup \gamma_2$: player 1 chooses to play γ_1 or γ_2,
- γ^*: γ is repeatedly played a finite number of times, player 1 chooses when to stop.
- γ^d: players switch roles.

We can define

- $\alpha \cap \beta \stackrel{def}{=} (\alpha^d \cup \beta^d)^d$ (demonic choice, choice for player 2)
- $\alpha^\times \stackrel{def}{=} ((\alpha^d)^*)^d$ (demonic iteration, player 2 chooses when to stop)

Semantic Game Constructs

- There is 1-1 correspondence:

$$\frac{E_\gamma\colon S \to \mathcal{P}(\mathcal{P}(S)) \quad \text{(upward-closed nbhd function)}}{\hat{E}_\gamma\colon \mathcal{P}(S) \to \mathcal{P}(S) \quad \text{(monotonic predicate transformer)}}$$

given by transpose: $s \in \hat{E}_\gamma(X) \iff X \in E_\gamma(s)$.

Semantic Game Constructs

- There is 1-1 correspondence:

$$\frac{E_\gamma : S \to \mathcal{P}(\mathcal{P}(S)) \quad \text{(upward-closed nbhd function)}}{\hat{E}_\gamma : \mathcal{P}(S) \to \mathcal{P}(S) \quad \text{(monotonic predicate transformer)}}$$

  given by transpose: $s \in \hat{E}_\gamma(X) \iff X \in E_\gamma(s)$.
- Algebraic operations on monotonic neighbourhood functions $E_\gamma : S \to \mathcal{P}(\mathcal{P}(S))$ are conveniently described in terms of $\hat{E}_\gamma$:

$$\begin{aligned}
\hat{E}_{\gamma_1;\gamma_2}(X) &= \hat{E}_{\gamma_1}(\hat{E}_{\gamma_2}(X)) \\
\hat{E}_{\gamma_1 \cup \gamma_2}(X) &= \hat{E}_{\gamma_1}(X) \cup \hat{E}_{\gamma_2}(X) \\
\hat{E}_{\gamma^*}(X) &= \mu Y.X \cup \hat{E}_\gamma(Y) \ \text{ (LFP of } Y \mapsto X \cup \hat{E}_\gamma(Y)) \\
\hat{E}_{\gamma^d}(X) &= \overline{\hat{E}_\gamma(\overline{X})}, \ (i.e.\ X \in E_{\gamma^d}(s) \text{ iff } \overline{X} \notin E_\gamma(s))
\end{aligned}$$

TU Delft

Game Logic Semantics

Interpretation of formulas and games in a GL model is defined by mutual induction:

- Interpretation of formulas (as usual):

$$M, x \models \langle\gamma\rangle\varphi \quad \text{iff} \quad [\![\varphi]\!]_M \in E_\gamma(x)$$

- Interpretation of games (as on previous slide) and for tests:

$$\hat{E}_{\varphi?}(X) = [\![\varphi]\!]_M \cap X$$

GL models are special multi-modal monotonic neighbourhood models.

Game Logic Axiomatisation

Let **GL** be the smallest set of formulas which contains

- monotonic modal logic $\mathbf{M}(\langle\gamma\rangle)$ for all $\gamma \in \Gamma$
- the following axiom schemas

$$\langle\gamma;\delta\rangle\varphi \leftrightarrow \langle\gamma\rangle\langle\delta\rangle\varphi \qquad \langle\gamma\cup\delta\rangle\varphi \leftrightarrow \langle\gamma\rangle\varphi \vee \langle\delta\rangle\varphi$$

$$\langle\psi?\rangle\varphi \leftrightarrow (\psi \wedge \varphi) \qquad \langle\gamma^{\mathsf{d}}\rangle\varphi \leftrightarrow \neg\langle\gamma\rangle\neg\varphi$$

$$\varphi \vee \langle\gamma\rangle\langle\gamma^*\rangle\varphi \to \langle\gamma^*\rangle\varphi$$

- and is closed under modus ponens and $\dfrac{\varphi \vee \langle\gamma\rangle\varphi \to \psi}{\langle\gamma^*\rangle\varphi \to \psi}$

Theorem (Parikh, 1985): **GL** without dual is sound and complete wr.t. dual-free GL models.

Theorem (Pauly, 2001): **GL** without iteration is sound and strongly complete w.r.t iteration-free GL models.

Open question: Completeness of full GL.

Complexity of Decision Procedures

Theorem (Pauly): Game Logic satisfiability is decidable in EXPTIME.

Theorem (Pauly): Model checking for Game Logic is polynomial for bounded alternation depth.

Game Logic References

- R. Parikh (1985): *The logic of games and its applications.* In: Topics in the Theory of Computation, Annals of Discrete Mathematics 14, Elsevier.
- M. Pauly & R. Parikh (2003): *Game Logic: An Overview.* Studia Logica 75(2), pp. 165-182.

- Extensions/variarions: stochastic GL (Doberkat), differential GL (Platzer), Multi-Agent Strategy Logic (Van Eijck),...

Concluding Remarks

Current Research: A unifying (coalgebraic) framework for dynamic logics like PDL, Game Logic and Coalition Logic.

- PDL-frame $S \to \mathcal{P}(S)^A$ where $\mathcal{P}$ is the powerset functor.
- GL-frame $S \to \mathcal{M}(S)^A$ where $\mathcal{M}$ is the monotone neighbourhood functor
- Both $\mathcal{P}$ and $\mathcal{M}$ are monads:
  sequential composition = Kleisli composition,
  skip program = unit of monad.
- A general framework for developing dynamic logics for structure $S \to T(S)^A$ parametric in the type T.
- Challenges: reduction axioms, identify algebraic "program/game" structure, how to prove completeness with fixpoint operators (cf. modal μ-calculus).