Team-based logics over Kripke structures

Heribert Vollmer
Outline of talk

Modal dependence logic
  Expressivity
  Complexity

Modal independence logic
  An example
  Expressivity
  Complexity

Extended modal dependence logic
  Expressivity
  Complexity

Future research directions
Syntax of ML:

\[ \varphi ::= p | \neg p | (\varphi \land \varphi) | (\varphi \lor \varphi) | \lozenge \varphi | \Box \varphi, \]

where \( p \) is an atomic proposition.
A team is a set of worlds.

\(K\) – a Kripke structure,
\(T\) – a team.

\(K, T \models p\) if \(p\) is true on every \(w \in T\).

\(K, T \models \neg p\) if \(\neg p\) is true on every \(w \in T\).

\(K, T \models \varphi_1 \lor \varphi_2\) if there exists a partition of \(T\) into \(T_1, T_2\) s.t. \(K, T_1 \models \varphi_1\) and \(K, T_2 \models \varphi_2\).

\(K, T \models \varphi_1 \land \varphi_2\) if \(K, T \models \varphi_1\) and \(K, T \models \varphi_2\).

\(K, T \models \Box \varphi\) if there exists a team \(T'\) that contains a successor world for every world in \(T\) s.t. \(K, T' \models \varphi\).

\(K, T \models \Diamond \varphi\) if for the team \(T'\) that contains all successor worlds of all worlds in \(T\) holds \(K, T' \models \varphi\).
Modal Dependence Logic

Syntax of MDL:

\[ \varphi ::= p \mid \neg p \mid =({\{p,}\} p) \mid \neg=({\{p,}\} p) \mid (\varphi \land \varphi) \mid (\varphi \lor \varphi) \mid \Diamond \varphi \mid \Box \varphi, \]

where \( p \) is an atomic proposition.

Semantics of MDL:

\[ K, T \models =({p_1, \ldots, p_n}) \text{ if for all } w_1, w_2 \in T, \]
\[ \text{if } \bigwedge_{i=1}^{n-1} (K, \{w_1\} \models p_i \iff K, \{w_2\} \models p_i) \]
\[ \text{then } K, \{w_1\} \models p_n \iff K, \{w_2\} \models p_n \]

\[ K, T \models \neg=({p_1, \ldots, p_n}) \text{ if } T = \emptyset \]
Independence Friendly Modal Logic

Caveat:

In FO logic, dependencies hold among FO-variables.

In MDL, dependencies hold among propositional variables.

Independence friendly modal logic: $\Box_1(\Diamond_2/\Box_1)p$

Several competing formalisms, cf., e.g., [Sevenster, Tulenheimo 07].
Expressivity of Modal Dependence Logic

MDL is more expressive than ML.

Proof. Closure under union.

On singleton teams of evaluation, MDL is as expressive as ML.

Proof.

- Use existentially quantified (Boolean) Skolem functions to replace dependence atoms.
- Replace quantifier by big disjunction over all possible functions.
- Over singleton teams, disjunction is the same as splitjunction.

[Sevenster 09]
Classical (Intuitionistic?) Disjunction

Extend MDL by:

\[ \varphi ::= p \mid \neg p \mid \varphi_1 \varphi_2 \mid (\{p, \neg p\}) \mid \neg (\{p, \neg p\}) \mid \\
(\varphi \land \varphi) \mid (\varphi \lor \varphi) \mid (\varphi \oplus \varphi) \mid \Diamond \varphi \mid \Box \varphi \]

\[ K, T \models \varphi_1 \ominus \varphi_2 \text{ if } K, T \models \varphi_1 \text{ or } K, T \models \varphi_2. \]

If we add disjunction to both logics, ML and MDL are of the same expressive power:

\[ ML < MDL < ML(\ominus) \equiv MDL(\ominus). \]
Satisfiability for Modal Dependence Logic

Sevenster 09:
Satisfiability for MDL (and MDL(⨁)) is NEXPTIME-complete.

Proof.
Upper bound: Express dependencies by Boolean Skolem functions. Use nondeterminism to guess functions.

Lower bound:
Reduction from Dependence-QBF which is NEXPTIME-complete [Peterson, Reif, Azhar 01].
## Satisfiability for Modal Dependence Logic

<table>
<thead>
<tr>
<th>□</th>
<th>◊</th>
<th>∧</th>
<th>∨</th>
<th>¬</th>
<th>T</th>
<th>⊥</th>
<th>=()</th>
<th>⊗</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>*</td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>+</td>
<td>*</td>
<td>NEXPTIME</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>-</td>
<td>*</td>
<td>PSPACE</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>PSPACE</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>-</td>
<td>+</td>
<td>$\Sigma_2^p$</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>+</td>
<td>*</td>
<td>+</td>
<td>$\Sigma_2^p$</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>-</td>
<td>-</td>
<td>coNP</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>*</td>
<td>+</td>
<td>*</td>
<td>-</td>
<td>coNP</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>NP</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>NP</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>NP</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>NP</td>
</tr>
<tr>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>NP</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>P</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
<td>+</td>
<td>*</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>P</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>P</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>P</td>
</tr>
<tr>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td>*</td>
<td>trivial</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>NP</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>+</td>
<td>*</td>
<td>+</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>+</td>
<td>NP</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>*</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>-</td>
<td></td>
<td>P</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>-</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td></td>
<td>P</td>
</tr>
</tbody>
</table>

10
The Complexity of Poor Man’s Logic

Poor man’s formulas = formulas that do not contain $\lor$.

**Hemaspaandra 01:** Poor man’s modal logic is PSPACE-complete over the class of Kripke structures in which every world has at most two successors.

**Proof.** Reduction from QBF: Express QBF-tree by alternations of modalities.

**Important:** Without disjunction, we cannot express tree-structure. Satisfiability for poor man’s modal logic over $K$ is coNP-complete.
The Complexity of Poor Man’s Logic

Lohmann, Vollmer 10:
Poor man’s modal dependence logic is NEXPTIME-complete.

Proof.

- Reduction from Dependence-QBF.
- Express QBF tree by alternations of modalities.
- Use dependence atoms to ensure that everything in the structure that does not belong to the tree is essentially nothing else than a copy of a subtree.
- Express dependencies by dependence atoms.

Note: Without negation or without dependence atoms, the complexity drops to $\Sigma_2^P$ (essentially using Ladner’s algorithm).
Modal Independence Logic

Syntax of MIL:

\[ \varphi ::= p | \neg p | \vec{p} \perp \vec{r} \vec{q} | (\varphi \land \varphi) | (\varphi \lor \varphi) | \lozenge \varphi | \square \varphi, \]

where \( p \) is an atomic proposition and \( \vec{p}, \vec{q}, \vec{r} \) are sets of atomic propositions.

Semantics of MIL:

\[ K, T \models \vec{p} \perp \vec{r} \vec{q} \text{ if for all } w_1, w_2 \in T \text{ that agree on } \vec{r} \text{ there exists } w_3 \in T \text{ that agrees with } w_1 \text{ on } \vec{p} \vec{r} \text{ and with } w_2 \text{ on } \vec{q}. \]
The Dining Cryptographers

Situation:
$n$ cryptographers sit around a table in a restaurant. After dinner they want to pay, but it turns out the bill has already been settled. Obviously only two possibilities:

(I) One of them has secretly paid and does not want to be identified.

(II) The NSA paid.

Task:
Design protocol to find out if (I) or (II) is the case, but without revealing the paying cryptographer in possibility I.
The Protocol

\(c_0, \ldots, c_{n-1}\): cryptographers, \(n \geq 3\) (in what follows, all indices modulo \(n\)).

- Let \(p_i := \begin{cases} 1 & \text{if } c_i \text{ paid} \\ 0 & \text{otherwise} \end{cases}\)

  Each cryptographer \(c_i\) knows \(p_i\) but not \(p_j\) for \(j \neq i\). There is at most one value \(i\) for which \(p_i = 1\). The goal of the protocol is to let everyone know the value \(P := p_0 \oplus p_1 \oplus \cdots \oplus p_{n-1}\).

- Each pair of adjacent cryptographers \(\{c_i, c_{i+1}\}\) agrees on a random bit \(\text{bit}_{\{i, i+1\}}\).

- Each cryptographer \(c_i\) publically announces the value \(\text{announce}_i = p_i \oplus \text{bit}_{\{i, i-1\}} \oplus \text{bit}_{\{i, i+1\}}\).

- It clearly follows that \(P = \text{announce}_0 \oplus \cdots \oplus \text{announce}_{n-1}\).
The Protocol as Kripke Structure
Correctness and Security

The protocol is correct: Everyone knows the result in the end.

Still to prove:
Anonymity requirement: After the protocol run, no cryptographer \( c_i \) should know more about the values \( p_j \) for \( j \neq i \) except for what follows from the values \( p_i \) (clearly, if \( c_1 \) paid then \( c_2 \) did not) or the computed result (if the NSA paid then no \( c_i \) payed).

Construction of formula \( \varphi \) that expresses the anonymity requirement.
Security Expressed in MIL

Let \( k_i := \{ p_i, \text{bit}_{i,i-1}, \text{bit}_{i,i+1} \} \cup \{ \text{announce}_j \mid j \neq i \} \), the set of variables \( c_i \) knows.

\( \varphi_g \) ensures that each individual bit that \( c_i \) knows does not tell whether \( c_j \) paid (unless \( i = j \) or \( p_i = 1 \)) by stipulating existence of red states in which all legal combinations appear:

\[
\bigwedge_{v \in k_i \setminus \{ p_i \}, \ k \in \{0,\ldots,n-1\}} \Diamond \Diamond (v \land p_k) \land \Diamond \Diamond (v \land \overline{p_k}) \land \Diamond \Diamond (\overline{v} \land p_k) \land \Diamond \Diamond (\overline{v} \land \overline{p_k})
\]

\[
\land \bigwedge_{i \neq j} \Diamond \Diamond (p_i \land \overline{p_j}) \land \Diamond \Diamond (\overline{p_i} \land p_j) \land \Diamond \Diamond (\overline{p_i} \land \overline{p_j})
\]
Security Expressed in MIL

Enumerate $\vec{k}_i = \{p_i, \text{bit}_{i,i-1}, \text{bit}_{i,i+1}\} \cup \{\text{announce}_j \mid j \neq i\}$ as $\vec{k}_i = \{v^i_1, \ldots, v^i_{n+2}\}$.

Let $V^{i}_{j \rightarrow k} := \{v^i_j, \ldots, v^i_k\}$ and $V^j := V^i_{j \rightarrow j}$.

Express that a single variable does not tell $c_i$ anything about the value of $p_k$, and their combination does not, either:

$$\varphi^{i,k} := \square \square ((V^i_{1 \perp p_k} V^i_2) \land (V^i_{1 \rightarrow 2 \perp p_k} V^i_3) \land \cdots \land (V^i_{1 \rightarrow n+1 \perp p_k} V^i_{n+2}))$$

$\varphi$ is the conjunction of $\varphi_g$ and all these formulas (and a few more . . .).
Expressivity of Modal Independence Logic

$\text{ML} \prec \text{MDL} \prec \text{MIL}$. 

Proof. Closure properties.

On singleton teams of evaluation, MIL is as expressive as ML.

Proof. MIL-formula on singletons captures a property of Kripke models that is

- invariant under modal bisimulation,
- only depends on the worlds that can be reached in a number of steps bounded by the modal depth of the formula.

Construct ML-formula that describes (by a big disjunction) all possibilities.

Note: Can be generalized to other generalized dependence atoms.
Satisfiability for Modal Independence Logic

Kontinen, Müller, Schnoor, Vollmer 14:
Satisfiability for MIL is \( \text{NEXPTIME} \)-complete.

Upper bound.
Embed MIL into the Gödel-Kalmár-Schütte fragment of all FO-sentences with prefix \( \exists^* \forall^2 \exists^* \) (without function symbols, without equality) in a satisfiability preserving way. Fragment is decidable in \( \text{NEXPTIME} \).

Note: Yields upper bound for MDL satisfiability as corollary, but uses a different proof.
Previous upper bound can be generalized to any extension of $\text{ML}$ by dependence atoms that can be defined in $\forall^2\exists^*$, for example:
inclusion, exclusion, …

Translation into Gödel-Kalmár-Schütte fragment also for strict semantics.

Example [Hella, Meier, Vollmer 14]:
Satisfiability for $\text{ML}(\subseteq)$ under strict semantics is in $\text{NEXPTIME}$ (in fact, complete).

On the other hand:
Satisfiability for $\text{ML}(\subseteq)$ under lax semantics is in $\text{EXPTIME}$. 
Extended Modal Dependence Logic

Syntax of EMDL:

$$\varphi ::= p \mid \neg p \mid = (\{\psi, \} \psi) \mid \neg = (\{\psi, \} \psi) \mid (\varphi \land \varphi) \mid (\varphi \lor \varphi) \mid \Diamond \varphi \mid \Box \varphi,$$

where $p$ is an atomic proposition and $\psi$ is an ML-formula.

Semantics of MDL:

$$K, T \models = (\psi_1, \ldots, \psi_n)$$ if for all $w_1, w_2 \in T$,

$$\text{if } \bigwedge_{i=1}^{n-1} (K, \{w_1\} \models \psi_i \iff K, \{w_2\} \models \psi_i)$$

$$\text{then } (K, \{w_1\} \models \psi_n \iff K, \{w_2\} \models \psi_n)$$

$$K, T \models \neg = (\psi_1, \ldots, \psi_n)$$ if $T = \emptyset$, 


Towards a Temporal Dependence Logic

Example:

\(= (\lozenge p, \lozenge^2 p, \ldots, \lozenge^n p, p)\)

“Truth of \(p\) at this moment only depends on the truth of \(p\) in the previous \(n\) time steps” (on suitable frame classes).

EMDL can be seen as a basic temporal dependence logic.
A Multimodal Example: Cellular Automata

Proposition $q$ describes state of a cell

$P$ is the past oriented successor relation in time
$L, R, U, D$ are the neighbour relations on the grid

$$= (⟨P⟩q, ⟨P⟩⟨L⟩q, ⟨P⟩⟨R⟩q, ⟨P⟩⟨U⟩q, ⟨P⟩⟨D⟩q, q, q)$$

expresses that

“The state of a cell at a given point in time is completely determined by its own state and the state of its neighbours at the previous time step.”
Expressivity of Extended Modal Dependence Logic

Ebbing, Hella, Meier, Müller, Virtema, Vollmer 13: EMDL is strictly more expressive than MDL.

Proof.

\((\Diamond p)\) is not expressible in MDL.

- Atomic propositions \(\Phi = \{p\}\) only
- Thus only dependences \(\eta = (p, \ldots, p)\).
- On \(K\), all MDL formulas are expressible in ML (by replacing \(\eta = (p, \ldots, p)\) by \(p\)).
- **Assumption:** \(\psi \in \text{MDL equivalent to } \eta = (\Diamond p)\).
- \(K, \{a\} \models = (\Diamond p)\) and \(K, \{b\} \models = (\Diamond p)\).
- By the assumption: \(K, \{a, b\} \models = (\Diamond p)\). 

\[\square\]
Expressivity of Extended Modal Dependence Logic

$$\text{MDL} < \text{EMDL} \equiv \text{ML}(\otimes).$$

**Note:** Translation from $\text{ML}(\otimes)$ into EMDL requires exponential blow-up.

Uses the new idea of the dimension of a formula [Luosto 14].

As above, on singletons all logics have the same expressive power.
Satisfiability for Extended Modal Dependence Logic

Ebbing, Hella, Meier, Müller, Virtema, Vollmer 13:
Satisfiability for EMDL is NEXPTIME-complete.

Upper bound.
Embed EMDL into MDL a satisfiability preserving way.
Questions

- Systematic *complexity* study of team based modal logic
  
  with generalized dependence atoms
  
  with generalized team connectives, e.g., intuitionistic implication
  
  over different frame classes
  
  comparing lax vs. strict semantics

- General theory of *expressivity* of different dependence atoms in modal context

- Temporal Dependence Logic?