Taming the Complexity of Temporal Epistemic Reasoning

Clare Dixon, Michael Fisher and Boris Konev

Department of Computer Science
University of Liverpool
Liverpool
UK

{clare,michael,konev}@csc.liv.ac.uk
www.csc.liv.ac.uk/∼{clare,michael,konev}

Motivation

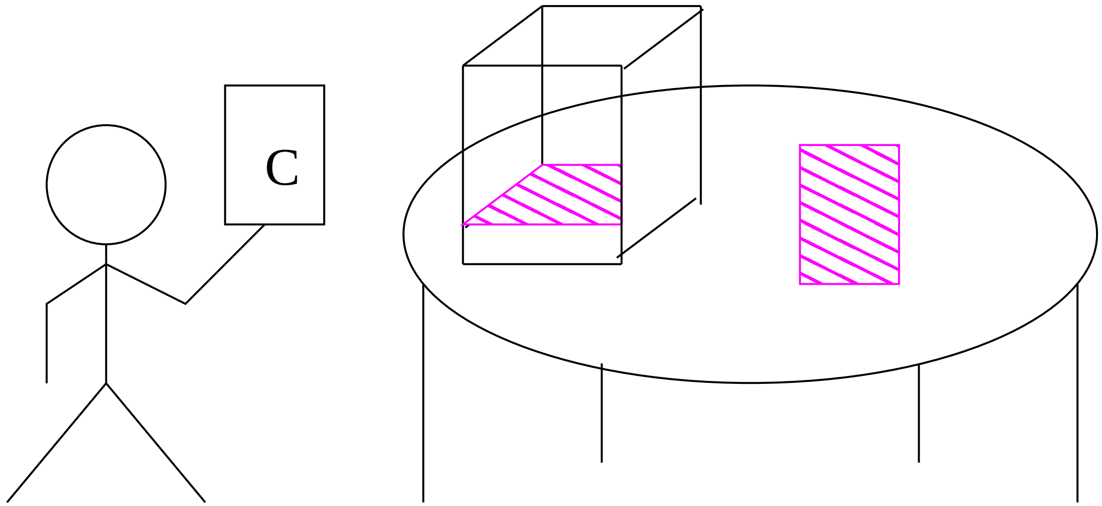
- Temporal logic of knowledge (TLK) has been used to represent and reason about how knowledge changes over time.
- The complexity of satisfiability for this logic is expensive, being PSPACE-complete.
- Often underlying such systems are sets of propositions where "exactly one" of each set holds at any state.
- We could try represent these directly in the specification however, these additional formulae lengthen and complicate the specification and adversely affect the performance of provers.
- Further, we would need some sort of universal operator to do this.

Overview and Contributions

- Here we consider the logic XL5, which is TLK but which allows a number of “exactly one” sets as input.
- This extends our work on temporal logics.
- The resulting logic allows more succinct specifications and simpler decision procedures (reducing certain aspects from *exponential* to *polynomial*).
- In this talk we:-
  - define an “exactly-one” temporal logic of knowledge;
  - provide a complete tableau calculus for this new logic;
  - consider the computational complexity of the tableau calculus; and
  - explore potential applications of the approach.

Motivating Example

Consider the card game† where there are three cards (hearts, clubs, spades) which may be on the table, in the card holder, or held by Wiebe.



Let $clubs_i$ where $i = w, h, t$ denote "Wiebe holds clubs" or "clubs is in the holder" or "clubs is on the table" respectively (and similarly for spades and hearts).

† H. van Ditmarsch, W. van der Hoek, and B. Kooi. Playing Cards with Hintikka — An Introduction to Dynamic Epistemic Logic. *Australasian Journal of Logic*, 3:108–134, 2005.

Specifying the Card Game I

Using a standard TLK, we would be forced to specify much background information. For example:

- Wiebe's card is spades or hearts or clubs:
  $(spades_w \vee clubs_w \vee hearts_w)$
- but Wiebe cannot hold both spades and clubs, both spades and hearts, or both clubs and spades:

$$\neg(spades_w \wedge clubs_w) \wedge \neg(spades_w \wedge hearts_w) \wedge \neg(clubs_w \wedge hearts_w).$$

- Similarly for the *holder* and the *table*.
- And Wiebe knows the above, e.g:
  $K_w(spades_h \vee clubs_h \vee hearts_h)$

Specifying the Card Game II

- The spades card must be either held by Wiebe or be in the holder or be on the table:
  $(spades_w \vee spades_h \vee spades_t)$
- but cannot be in more than one place:

$$\neg(spades_w \wedge spades_h) \wedge \neg(spades_w \wedge spades_t) \wedge \neg(spades_h \wedge spades_t).$$

- Similarly for both the *hearts* and *clubs* cards.
- And again Wiebe knows the above, e.g:
  $K_w(spades_w \vee spades_h \vee spades_t)$
- All the above statements hold globally.

The Logic XL5

- The syntax and semantics of "XL5" are essentially that of a propositional temporal logic of knowledge (a *fusion* of propositional, linear, discrete, temporal logic and S5 modal logic of knowledge).
- Formulae of XL5($\mathcal{P}^1, \mathcal{P}^2, \ldots$) are constructed under the restrictions that *exactly* one proposition from every set $\mathcal{P}^i$ is true in every state.
- Also there exists a set of unconstrained propositions, $\mathcal{A}$.
- Thus, XL5() is a standard propositional, linear temporal logic of knowledge, while XL5($\mathcal{P}$,$\mathcal{Q}$,$\mathcal{R}$) has models where exactly one of each of $\mathcal{P}$, $\mathcal{Q}$, and $\mathcal{R}$ must hold at every moment.

Syntax

The formulae of XL5($\mathcal{P}^1, \mathcal{P}^2, \ldots, \mathcal{P}^m$) over a set of agents $Ag = \{1, \ldots, n\}$ are constructed using:-

- a set $\mathcal{P}^1 \cup \mathcal{P}^2 \cup \ldots \cup \mathcal{P}^m \cup \mathcal{A} =$ PROP of proposition symbols and the constants F and T;
- the connectives $\neg$, $\vee$, $\bigcirc$, $\mathcal{U}$ and K_i (where $i \in Ag$).

Well-formed formulae (WFF) of XL5($\mathcal{P}^1, \mathcal{P}^2, \ldots, \mathcal{P}^m$)

- F and T and any element of PROP is in WFF;
- if A and B are in WFF and $i \in Ag$ then so are

$$\neg A \qquad A \vee B \qquad K_i A \qquad A \,\mathcal{U}\, B \qquad \bigcirc A.$$

The operators $\wedge$, $\Rightarrow$, $\Diamond$ and $\Box$ are defined as equivalences.

Semantics I

A *timeline*, t, is an infinitely long, linear, discrete sequence of states, indexed by the natural numbers.

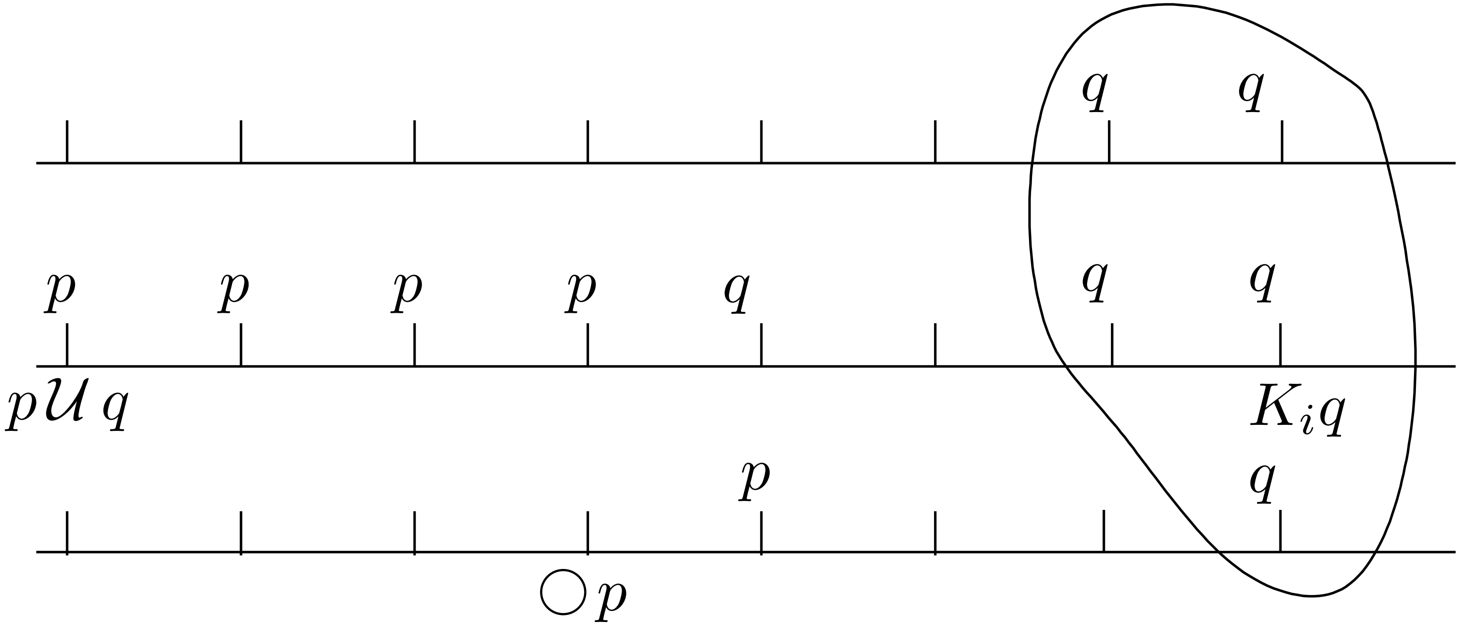
A point is a pair (t, u), where t is a timeline and $u \in \mathbb{N}$.

A *model*, M, for KL$_n$ is a structure $M = \langle TL, R_1, \ldots, R_n, \pi \rangle$:

- TL is a set of timelines;
- $R_i \subseteq Points \times Points$ is the agent accessibility relation where each R_i is an equivalence relation;
- π is a valuation ($\pi : Points \times$ PROP $\rightarrow \{T, F\}$) which satisfies the “exactly one” sets.

For any formula A, if there is *some* model M and timeline t such that $\langle M, (t, 0) \rangle \models A$, then A is said to be *satisfiable* (respectively for *all* models then A is said to be *valid*).

Semantics II



Where the Booleans have the usual semantics.

Also $\Diamond$ and $\Box$ are defined as equivalences, i.e. $\Diamond p \equiv T\,\mathcal{U}\,p$ and $\Box p \equiv \neg\Diamond\neg p$.

Complexity of XL5

Theorem 1 *The satisfiability problem for* XL5$(\mathcal{P})$*, even if all variables belong to the single constrained set* $\mathcal{P}$*, is PSPACE-complete.*

Theorem 2 *The satisfiability problem for following two fragments of* XL5$(\mathcal{P})$ *is NP-hard:*

- *all variables belong to the single constrained set* $\mathcal{P}$*, there is one agent and temporal operators are not used; and*
- *all variables belong to the single constrained set* $\mathcal{P}$ *and modal operators are not used.*

Later we show that XL5 reasoning is tractable if the number of occurrences of temporal and modal operators is bounded.

Overview of the Tableau Algorithm

- To show φ is satisfiable the tableau algorithm constructs sets of *extended assignments* (EA) of propositions and temporal and modal subformulae.
- Extended assignments are a mapping of these subformulae to true or false, that satisfy both the "exactly one" sets and φ.
- To achieve this we use a DPLL-based expansion rather than the usual alpha and beta rules.
- Next the algorithm attempts to satisfy modal formulae, of the form $\neg K_i \psi$, and temporal formulae, of the form $\bigcirc \psi$ and $\psi_1 \mathcal{U} \psi_2$ (or their negations), made true in such an extended assignment by constructing R_i and "next time" successors.

Extended Assignments (EA)

- Let φ be an XL5 formula, and
  - $\text{PROP}(\varphi)$ be the set of all propositions occurring in φ,
  - $\text{MOD}(\varphi)$ be the set of all modal subformulae of φ,
  - $\text{TEMP}(\varphi)$ be the set of all temporal subformulae of φ

  then an EA ν *for* φ is a mapping from $\text{PMT}(\varphi) = \text{PROP}(\varphi) \cup \text{MOD}(\varphi) \cup \text{TEMP}(\varphi)$ to $\{T, F\}$.
- Every EA ν can be represented by a set of formulae

$$\Delta_\nu = \bigcup_{\substack{\psi \in \text{PMT}(\varphi) \\ \nu(\psi) = T}} \{\psi\} \cup \bigcup_{\substack{\psi \in \text{PMT}(\varphi) \\ \nu(\psi) = F}} \{\neg\psi\}$$

Compatibility

Let ψ be a XL5 formula such that $\text{PMT}(\psi) \subseteq \text{PMT}(\varphi)$. An EA ν for φ is *compatible* with ψ if, and only if:

- For every set $\mathcal{P}^i$, there exists exactly one proposition $p \in \mathcal{P}^i$ such that $\nu(p) = T$.
- Replacing every occurrence of $\psi' \in \text{PMT}(\psi)$ such that ψ' is not in the scope of another modal or temporal operator in ψ, with $\nu(\psi')$, evaluates to T.
- If $\nu(K_j\chi) = T$, for some modal subformula $K_j\chi$ of ψ, then ν is compatible with χ.
- If $\nu(\chi_1 \mathcal{U} \chi_2) = T$, for some temporal subformula $\chi_1 \mathcal{U} \chi_2$ of ψ, then ν is compatible with χ_1 or χ_2.
- If $\nu(\chi_1 \mathcal{U} \chi_2) = F$, for some temporal subformula $\chi_1 \mathcal{U} \chi_2$ of ψ, then ν is compatible with $\neg\chi_2$.

Tableau Algorithm I

Let φ be a XL5 formula to be shown (un)satisfiable.

1. *Initialisation*. First, set $S = \eta = R_1 = \cdots = R_n = L = \emptyset$. Construct the set of all EAs for φ compatible with φ and add new states.
2. *Creating R_i successors*. For any state s such that $L(s) = \nu$ for each $\neg K_i\psi \in \Delta_{L(s)}$ let

$$\psi' = \neg\psi \wedge \bigwedge_{K_i\chi \in \Delta_{L(s)}} K_i\chi \wedge \chi \quad \wedge \bigwedge_{\neg K_i\chi \in \Delta_{L(s)}} \neg K_i\chi$$

   For each ψ' above construct the set of EAs for φ compatible with ψ' and add new states and relations R_i.

Tableau Algorithm II

3. *Creating* η *successors*. For any state s such that $L(s) = \nu$ create the set of formulae $next(\nu)$ where $next(\nu)$ is the smallest subset of Δ_ν such that:
   - $\bigcirc\chi \in \Delta_\nu$ then $\chi \in next(\nu)$;
   - $\neg\bigcirc\chi \in \Delta_\nu$ then $\neg\chi \in next(\nu)$;
   - $\chi_1 \mathcal{U} \chi_2 \in \Delta_\nu$ but ν is not compatible with χ_2, then $\chi_1 \mathcal{U} \chi_2 \in next(\nu)$; and
   - $\neg(\chi_1 \mathcal{U} \chi_2) \in \Delta_\nu$ but ν is not compatible with $\neg\chi_1$, then $\neg(\chi_1 \mathcal{U} \chi_2) \in next(\nu)$.

   Let ψ' be the conjunction of formulae in $next(\nu)$. For each ψ' construct the set of EAs for φ compatible with ψ' and add new states and η-relations.

Tableau Algorithm III

4. *Contraction*. Delete any state s with $L(s) = \nu$ where
   - there exists a formula $\neg K_i \chi \in \Delta_{L(s)}$ and there is no state $s' \in S$ such that $(s, s') \in R_i$ and $L(s')$ is compatible with $\neg\chi$,
   - $next(\nu)$ is not empty but there is no $s' \in S$ such that $(s, s') \in \eta$, or
   - there exists a formula $\chi_1 \mathcal{U} \chi_2 \in \Delta_{L(s)}$ and there is no $s' \in S$ such that $(s, s') \in \eta^*$ and $L(s')$ is compatible with χ_2 (η^* is the transitive reflexive closure of η).

   until no further deletions are possible.

   The tableau algorithm is *successful* iff, the structure contains a state s such that $L(s)$ is compatible with φ.

Example

Recall Wiebe's card game with three cards.

We add the following assumptions relating to time:

- originally Wiebe has been dealt the clubs card (but has not looked at the card so doesn't know this yet) $clubs_w$;
- at the next step Wiebe looks at his card so he knows that he has the *clubs* card, so $\bigcirc K_w clubs_w$.

We try show that in the next moment Wiebe doesn't hold the spades card.

$$(clubs_w \land \bigcirc K_w clubs_w) \Rightarrow \bigcirc \neg spades_w$$

Using Exactly One Sets

Instead of the background knowledge we specified earlier we can use

$$\mathsf{XL5}(\mathcal{P}^1, \mathcal{P}^2, \mathcal{P}^3, \mathcal{P}^4, \mathcal{P}^5, \mathcal{P}^6)$$

where

- $\mathcal{P}^1 = \{spades_w, clubs_w, hearts_w\}$
- $\mathcal{P}^2 = \{spades_h, clubs_h, hearts_h\}$
- $\mathcal{P}^3 = \{spades_t, clubs_t, hearts_t\}$
- $\mathcal{P}^4 = \{spades_w, spades_h, spades_t\}$
- $\mathcal{P}^5 = \{clubs_w, clubs_h, clubs_t\}$
- $\mathcal{P}^6 = \{hearts_w, hearts_h, hearts_t\}$

Applying the Tableau

$$\varphi = \neg((clubs_w \wedge \bigcirc K_w clubs_w) \Rightarrow \bigcirc \neg spades_w)$$

We construct the set of EAs for φ compatible with φ.

$$\mathcal{I}_0 = \{clubs_w, \bigcirc K_w clubs_w, \neg \bigcirc \neg spades_w\}$$

$$\begin{aligned}
\Delta_{L(s_0)} &= \mathcal{I}_0 \cup \{K_w clubs_w, hearts_h, spades_t\} \\
\Delta_{L(s_1)} &= \mathcal{I}_0 \cup \{K_w clubs_w, hearts_t, spades_h\} \\
\Delta_{L(s_2)} &= \mathcal{I}_0 \cup \{\neg K_w clubs_w, hearts_h, spades_t\} \\
\Delta_{L(s_3)} &= \mathcal{I}_0 \cup \{\neg K_w clubs_w, hearts_t, spades_h\}
\end{aligned}$$

Next we construct R_w successors to s_2 and s_3.

Constructing Next-Successors

Constructing η successors for s_0–s_3

$$next(L(s_i)) = \{K_w clubs_w, \neg\neg spades_w\}$$
and $\psi'' = K_w clubs_w \wedge \neg\neg spades_w$.

Let $\mathcal{I}_1 = \{K_w clubs_w, clubs_w, spades_w\}$

There are no EAs for φ which are compatible with ψ''. Any such EAs would contain $\mathcal{I}_1$ as a subset.

As s_0–s_3 have no η successors they are deleted.

As there is no remaining state compatible with φ the tableau is unsuccessful and so φ is unsatisfiable and $(clubs_w \wedge \bigcirc K_w clubs_w) \Rightarrow \bigcirc \neg spades_w$ is valid.

Correctness and Complexity

Theorem 3 *Let $\mathcal{P}^1$,..., $\mathcal{P}^m$ be sets of constrained propositions, and φ be an* XL5$(\mathcal{P}^1, \ldots, \mathcal{P}^m)$ *formula such that $\bigcup_{i=1}^m \mathcal{P}^i \subseteq$* PROP$(\varphi)$*. Then*

- *φ is satisfiable if, and only if, the tableau algorithm applied to φ returns a structure $(S, \eta, R_1, \ldots, R_n, L)$ in which there exists a state $s \in S$ such that $L(s)$ is compatible with φ.*
- *The tableau algorithm runs in time polynomial in $\left((k+t) \times |\mathcal{P}^1| \times \ldots \times |\mathcal{P}^m| \times 2^{|\mathcal{A}|+k+t}\right)$, where $|\mathcal{P}^i|$ is the size of the set $\mathcal{P}^i$ of constrained propositions, $|\mathcal{A}|$ is the size of the set $\mathcal{A}$ of non-constrained propositions, k is the number of modal operators in φ, and t is the number of temporal operators in φ.*

Potential Application Areas

- Distributed Systems
- Learning and Knowledge Evolution
- Security
- Robotics
- Planning and Knowledge Representation

Conclusions

- We have defined a temporal logic of knowledge which allows “exactly one” constraints to be defined as parameters.
- We have motivated the need for such constraints by considering a number of application areas.
- We have provided a tableau based algorithm to prove XL5 formulae which replaces the usual alpha and beta rules with a DPLL-based expansion.
- We analysed its complexity which shows that the tableau is useful when applied to problems with a large number of constrained propositions and a comparatively low number of unconstrained propositions, modal and temporal operators in the formula to be proved.