MORE ABOUT MODAL LOGIC

1. Finite model property
2. Decidability
3. Translation to predicate logic
4. Multi-modal logics
5. Computational complexity

M. Fitting: *Basic Modal Logic*, Sections 1.10, 1.12 ja 1.14 (pp. 403 – 405, 408 – 410, and 416 – 419).

1. Finite Model Property

Decidability

- Next we consider computational properties of modal logics. As usual we use as the formal model of computation Turing machines. Hence, when saying that there is an algorithm computing/solving some problem we mean that there is a Turing machine capable of computing/solving the problem.
- We say that logic $\mathbf{L}$ is decidable, if there is an algorithm such that given a formula as input it decides whether the formula is $\mathbf{L}$-valid or not.
- Such an algorithm is called a decision procedure for $\mathbf{L}$.

Semi-Decidability

- If there is a proof system (for example Hilbert-style proof theory) for logic $\mathbf{L}$, then $\mathbf{L}$ is semi-decidable: there is an algorithm that halts and says “yes” if the formula given as input is $\mathbf{L}$-valid (but might not halt if the formula is not $\mathbf{L}$-valid).

  Proof sketch: the algorithm starts enumerating $\mathbf{L}$-proofs in some systematic way and halts if a proof for the formula is found.

$\Longrightarrow$ The set of $\mathbf{L}$-valid formulas is recursively enumerable.

Finite Model Property

Definition. Modal logic $\mathbf{L}$ has the finite model property if for every formula P which is not $\mathbf{L}$-valid, there is a finite $\mathbf{L}$-model that has a world where P is false.

- If a logic $\mathbf{L}$ has the finite model property, then the set of formulas that are not $\mathbf{L}$-valid is semi-decidable.

  Proof sketch. Enumerate in some systematic way finite models from small models to bigger ones and check for each of them whether the model has a world where P is false.
- Hence, if a modal logic $\mathbf{L}$ has a proof system and the finite model property, then $\mathbf{L}$ is decidable.
- How can we show that a logic has the finite model property?

$\Longrightarrow$ Filtration.

Example. Filtration for Modal Logic K

Theorem. Modal logic $\mathbf{K}$ has the finite model property.

Proof. Assume that a formula Q is not $\mathbf{K}$-valid. Hence, there is a model $\mathcal{M} = \langle S, R, v\rangle$ and a world $s_0 \in S$ such that $\mathcal{M}, s_0 \nVdash Q$.

We construct a finite (quotient) model $|\mathcal{M}|$ from $\mathcal{M}$ by filtration.

Let $\mathrm{Sub}(Q)$ be the set of all subformulas of Q.

- Define an equivalence relation $\sim$ for the set S: $s \sim t$ if for all $P \in \mathrm{Sub}(Q)$, $\mathcal{M}, s \Vdash P$ iff $\mathcal{M}, t \Vdash P$.
- Let $|s| = \{t \in S : t \sim s\}$ and $|S| = \{|s| : s \in S\}$. Now $|S|$ is finite because it has at most 2^n elements where n is the number of subformulas of Q: every $|s| \in |S|$ is a set of worlds such that exactly the same subset of subformulas of Q are true in each world in $|s|$.

Filtration—cont'd

Consider now a model $|\mathcal{M}| = \langle |S|, |R|, |v|\rangle$ where the relation $|R|$ is defined as follows:

$$|s||R||t| \text{ iff there are elements } s' \in |s| \text{ and } t' \in |t| \text{ such that } s'Rt'$$

and the valuation $|v|$: $|v|(|s|, P) = v(s, P)$.

We show by structural induction that for every $P \in \mathrm{Sub}(Q)$

$$\mathcal{M}, s \Vdash P \text{ iff } |\mathcal{M}|, |s| \Vdash P.$$

- P is an atomic proposition: $|v|(|s|, P) = v(s, P)$.
- $\neg P$: $\mathcal{M}, s \Vdash \neg P$ iff $\mathcal{M}, s \nVdash P$ iff (IH) $|\mathcal{M}|, |s| \nVdash P$ iff $|\mathcal{M}|, |s| \Vdash \neg P$.
- $P \to Q$: can be proved in a similar way.

Filtration—cont'd

- $\Box P$: ($\Leftarrow$) Let $\mathcal{M}, s \nVdash \Box P$. Then there is a world t such that sRt and $\mathcal{M}, t \nVdash P$. Hence, $|s||R||t|$ and $|\mathcal{M}|, |t| \nVdash P$ (IH) and $|\mathcal{M}|, |s| \nVdash \Box P$.
- $\Box P$: ($\Rightarrow$) Let $|\mathcal{M}|, |s| \nVdash \Box P$ hold. Then there is $|t|$ such that $|s||R||t|$ and $|\mathcal{M}|, |t| \nVdash P$. Hence, $\mathcal{M}, t \nVdash P$ (by IH). Thus, there are worlds $s' \in |s|$ and $t' \in |t|$ such that $s'Rt'$ and $\mathcal{M}, t' \nVdash P$. So $\mathcal{M}, s' \nVdash \Box P$ and $\mathcal{M}, s \nVdash \Box P$.

The theorem follows from the result above:

- As $\mathcal{M}, s_0 \nVdash Q$ and $Q \in \mathrm{Sub}(Q)$, $|\mathcal{M}|, |s_0| \nVdash Q$ holds.
- Hence, logic $\mathbf{K}$ has the finite model property.

Many other normal modal logics have the finite model property, for example, $\mathbf{T}, \mathbf{K4}, \mathbf{S4}, \mathbf{KB}, \mathbf{B}$, $\mathbf{S5}, \mathbf{D}, \mathbf{D4}, \mathbf{DB}, \mathbf{KD45}$.

2. Decidability

- If we can give an upper bound on the size of the counter-model, the logic is decidable.
- This observation does not lead to a very efficient decision procedure.
- The tableau method provides a more efficient approach.

Example. We can show that logic $\mathbf{K}$ is decidable by showing that the tableau method for $\mathbf{K}$ provides a decision procedure that halts on every formula.

For this argument we need König's lemma:

Lemma. If a tree has an infinite number of nodes but every node has a finite number of child nodes, then the tree has an infinite branch.

A decision procedure deciding whether a formula P is $\mathbf{K}$-valid:

1. Take as the root of the tableau $\langle 1\rangle\neg P$.
2. Until the tableau is closed or all formulas have been marked used:

2.1 Choose the uppermost unused node σQ in the tableau.

2.2 If Q is not a literal, then for each open branch θ containing σQ do:

- If σQ is of the form $\sigma\neg\neg Q'$, extend θ by the node $\sigma Q'$.
- If σQ is of the form $\sigma\alpha$, extend θ by nodes $\sigma\alpha_1$ and $\sigma\alpha_2$.
- If σQ is of the form $\sigma\beta$, extend θ to two branches one containing $\sigma\beta_1$ and the other $\sigma\beta_2$.
- If σQ is of the form $\sigma\neg\Box P$, extend θ by $\sigma n\neg P$ with some σn which is unrestricted on θ and then with $\sigma n X$ for every $\sigma\Box X$ that appears on the branch θ.
- If σQ is of the $\sigma\Box P$, extend θ with $\sigma n P$ for every σn which is available on θ if $\sigma n P$ is not already contained in θ.

2.3 Mark σQ used.

Proposition. The procedure above halts for every formula P after a finite number of steps.

Proof. Assume that there is a formula P for which the procedure does not halt after a finite number of steps.

- Then the procedure constructs an infinite $\mathbf{K}$-tableau, i.e., a tree where each node has at most to child nodes.
- By König's lemma the tableau has an infinite branch θ.
- In θ there is an infinite number of prefixed formulas, which are all different (we are assuming that the procedure extends the branch with a prefixed formula only if it is not contained in the branch).
- For every prefix σ, if σQ occurs on a branch θ, then Q is a subformula of P or the negation of a subformula. As P has only a finite number of subformulas, each prefix can occur only a finite number of times.
- Hence, the branch θ must contain an infinite number of different prefixes.

Proof—cont'd

There are two possibilities:

(i) The branch θ has an infinite number of prefixes of some length n.

- Let n be the smallest natural number such that θ has infinitely many prefixes of length n.
- Now it must be the case that $n \neq 1$ because the only prefix of length one is $\langle 1\rangle$ and it occurs only finitely many times in θ.
- If $n > 1$, then the only way of producing a prefix of length n is to use $\neg\Box$ or $\Box$ rule to a formula σQ where σ is of length $n-1$.
- Hence, if the branch contains an infinite number of prefixes of length n, then it must have an infinite number of prefixes of length $(n-1)$, which is in contradiction with the choice of n.

Proof—cont'd

$\Longrightarrow$ Hence, possibility (i) cannot be the case and, thus, (ii) θ has a finite number of prefixes of length n for every natural n.

- As there is an infinite number of prefixes in θ , it must contain prefixes of infinitely many different lenghts.
- We show that this is impossible by establishing that for every formula σQ in θ the following property L holds:
  the sum of the length of σ with the number of modal operators in Q is always at most $1+m$, where m is the number modal operators in P.
- Clearly (L) holds for the root of the tableau $\langle 1\rangle\neg P$.
- If a formula is obtained by α or β rules, then (L) holds also for any new prefixed formula obtained by the rules.

Proof—cont'd

- If the formula has been obtained using the $\Box$ or $\neg\Box$ rule, it is of the form $\sigma k Q$ and it is derived from a formula of the form $\sigma Q'$. Assume that (L) hold for $\sigma Q'$. Now the length of the prefix σk is the length of the prefix $\sigma + 1$ but the formula Q has one modal operator less than the formula Q'. Hence, (L) holds also for $\sigma k Q$.
- Since (L) holds for every prefixed formula on the branch, the branch can contain only prefixes of the length at most $m+1$.

$\Longrightarrow$ Hence, the remaining possibility (ii) leads also to contradiction and, hence, the procedure halts on every formula P.

- A corresponding decision procedure works for modal logics where the transitivity of the frames is not required.
- For logics which assume transitivity a stronger halting condition is needed, i.e., a condition for stopping of expanding branches.

3. Translation to Predicate Logic

Modal logic can be translated to a restricted subset of predicate logic where we use

- for every atomic proposition P, a one-argument predicate symbol P;
- two variables x_1 and x_2;
  (notation: if x one of the variables, then x' is the other.)
- two-place predicate symbol R.

Definition. Let τ be a mapping from modal formulas and variables to formulas in predicate logic such that

1. $\tau(\top,x) = \top$; $\tau(\bot,x) = \bot$;
2. $\tau(P,x) = P(x)$ for atomic propositions P;
3. $\tau(\neg P,x) = \neg\tau(P,x)$;
4. $\tau(P \to Q,x) = \tau(P,x) \to \tau(Q,x)$;
5. $\tau(\Box P,x) = \forall x'(R(x,x') \to \tau(P,x'))$;

Example. $\tau(\neg\Box P \to \Box\neg\Box P, x_1) = \tau(\neg\Box P,x_1) \to \tau(\Box\neg\Box P,x_1)$
$= \neg\tau(\Box P,x_1) \to \forall x_2(R(x_1,x_2) \to \tau(\neg\Box P,x_2))$
$= \neg(\forall x_2(R(x_1,x_2) \to \tau(P,x_2))) \to \forall x_2(R(x_1,x_2) \to \neg\tau(\Box P,x_2))$
$= \neg(\forall x_2(R(x_1,x_2) \to P(x_2))) \to$
$\quad \forall x_2(R(x_1,x_2) \to \neg(\forall x_1(R(x_2,x_1) \to \tau(P,x_1))))$
$= \neg(\forall x_2(R(x_1,x_2) \to P(x_2))) \to$
$\quad \forall x_2(R(x_1,x_2) \to \neg(\forall x_1(R(x_2,x_1) \to P(x_1))))$.

Mapping τ Preserves Validity

Theorem. Let P be a modal formula.

1. P is $\mathbf{K}$-valid iff $\forall x_1 \tau(P,x_1)$ is valid in predicate logic.
2. $\Sigma \models_{\mathbf{K}} \emptyset \Longrightarrow P$ iff $\tau(\Sigma) \models_{\mathrm{cl}} \forall x_1\tau(P,x_1)$,
   where $\tau(\Sigma) = \{\forall x_1 \tau(Q,x_1) \mid Q \in \Sigma\}$ and
   $\models_{\mathrm{cl}}$ is the logical consequence relation in predicate logic.

- Modal logic $\mathbf{T}$:
  P is $\mathbf{T}$-valid iff $\{\forall x_1 R(x_1,x_1)\} \models_{\mathrm{cl}} \forall x_1\tau(P,x_1)$.
- Modal logic $\mathbf{S5}$:
  P is $\mathbf{S5}$-valid iff $\Sigma_{\mathbf{S5}} \models_{\mathrm{cl}} \forall x_1\tau(P,x_1)$
  where $\Sigma_{\mathbf{S5}} = \{\forall x_1 R(x_1,x_1), \forall x_1\forall x_2(R(x_1,x_2) \to R(x_2,x_1)),$
  $\forall x_1\forall x_2\forall x_3(R(x_1,x_2) \land R(x_2,x_3) \to R(x_1,x_3))\}$.

4. Multi-modal Logic

Example. Multi-agent logic of knowledge $\mathbf{S5_n}$:

- n agents and corresponding knowledge operators $K_i, i = 1, \ldots, n$. For example, formulas of the form $K_1 K_2 \neg K_1 P$ allowed.
- Models are tuples of the form $\langle S, R_1, \ldots, R_n, v \rangle$ and $\mathcal{M}, s \mid\!\mid\!\!- K_i P$ iff $\mathcal{M}, t \mid\!\mid\!\!- P$ for every $t \in S$ such that $s R_i t$.
- A (Hilbert-style) proof system consists of **S5** axioms for every K_i.
- Everybody knows (EP): $\mathcal{M}, s \mid\!\mid\!\!- EP$ iff $\mathcal{M}, s \mid\!\mid\!\!- K_i P$ for all $i = 1, \ldots, n$.
- Axiom: $EP \leftrightarrow K_1 P \wedge \cdots \wedge K_n P$.

Common Knowledge: CP

- $\mathcal{M}, s \mid\!\mid\!\!- CP$ iff $\mathcal{M}, s \mid\!\mid\!\!- E^k P$ for every $k = 1, 2, \ldots$

  $\Longrightarrow$ $\mathcal{M}, s \mid\!\mid\!\!- CP$ iff $\mathcal{M}, t \mid\!\mid\!\!- P$ for each $t \in S$ such that the world t is C-accessible from the world s (i.e., there is $k \geq 1$ and a sequence $(s =) t_0, t_1, \ldots, t_k (= t)$ where for every $j = 0, \ldots, k-1$, $t_j R_i t_{j+1}$ holds for some $i, 1 \leq i \leq n$.)
- Axiom: $CP \rightarrow E(P \wedge CP)$.
- Inference rule:

$$\frac{P \rightarrow E(Q \wedge P)}{P \rightarrow CQ}$$

- **Example.** $CP \rightarrow K_1 K_2 \cdots K_n P$ is $\mathbf{S5_n}$-valid

5. Computational Complexity

Different computational problems

- Model checking (Is a formula true in a model?)
- Satisfiability (Does a formula have a model where it is true?)
- Validity
- Logical consequence

We consider here model checking and satisfiability because

1. a formula P is valid iff $\neg P$ is not satisfiable;
2. $\{\} \models_{\mathbf{L}} \{Q_1, \ldots Q_n\} \Longrightarrow P$ iff $(Q_1 \wedge \cdots \wedge Q_n) \rightarrow P$ is valid iff $Q_1 \wedge \cdots \wedge Q_n \wedge \neg P$ is not satisfiable.

Computational Complexity of Model Checking

- Whether a formula is true in a given world of a model can be checked in polynomial time (w.r.t. the size of the model and length of the formula) for all modal logics we have considered so far.
- However, not every modal logic has this property (for example, many temporal logics do not).
- All problems solvable in polynomial time can be reduced to evaluating a propositional formula:

  Evaluating the truth value of a Boolean circuit is a **P**-complete problem!

Computational Complexity of Satisfiability

- Deciding satisfiability is **NP**-complete for propositional logic.
- All normal modal logics contain propositional logic as a special, so deciding satisfiability for them is **NP**-hard.
- For modal logics $\mathbf{S5}, \mathbf{KD45}$ deciding satisfiability is **NP**-complete.
  For these logics, non-valid formulas have small counter-models (the number of worlds is at most the number of subformulas which implies that deciding satisfiability is in **NP**).
- For modal logics $\mathbf{K}, \mathbf{T}, \mathbf{S4}$ the problem is **PSPACE**-complete:
  in these logics counter-models can be of exponential size.
- For logics $\mathbf{S5_n}, \mathbf{KD45_n}$ the problem is **PSPACE**-compete.
- For modal logics $\mathbf{S5_n^C}, \mathbf{KD45_n^C}$ the problem is **EXPTIME**-complete.

Summary

- Modal logics have varying computational properties.
- Decidability can often be established by providing a proof system and showing that the logic has the finite model property.
- More efficient decision procedures can be obtained using the tableau method.
- Many modal logics can be translated in a systematic way to a restricted subset of predicate logic.
- The one modal operator case can be generalized to the multi-modal case in the possible world semantics by introducing an accessibility relation to each of the modal operators.
- Modal logics differ in their inherent computational complexity.