Non-classical logics

Lecture 16 + 17: Modal logics (Part 4)

Viorica Sofronie-Stokkermans

sofronie@uni-koblenz.de

Until now

- History and Motivation
- Propositional modal logic

  Syntax

  Inference systems and proofs

  Semantics (models, validity, satisfiability)

  Entailment (local/global); Deduction theorem

  Correspondence theory; First-order definability

  Proof Systems

Theorem proving in modal logics

- Inference system — last time
- Tableau calculi
- Resolution

Theorem proving in modal logics

- Inference systems
- Tableau calculi today
- Resolution today

Modal logic

Theorem proving in modal logics

- Inference systems
- Tableau calculi
- Resolution

Tableau calculus

We use labelled formulae

TG standing for "Formula G is true"

FG standing for "Formula G is false"

Tableau calculus

Formula classes

α-Formulae	$T(A \wedge B)$, $F(A \vee B)$, $F(A \rightarrow B)$, $F(\neg A)$
β-Formulae	$T(A \vee B)$, $F(A \wedge B)$, $T(A \rightarrow B)$, $T(\neg A)$
ν-Formulae	$T\ \Box A$, $F\ \Diamond A$
π-Formulae	$T\ \Diamond A$, $F\ \Box A$

Tableau calculus

Successor formulae

α	α_1	α_2
$T(A \wedge B)$	TA	TB
$F(A \vee B)$	FA	FB
$F(A \rightarrow B)$	TA	FB
$F(\neg A)$	TA	TA

β	β_1	β_2
$T(A \vee B)$	TA	TB
$F(A \wedge B)$	FA	FB
$T(A \rightarrow B)$	TB	FA
$T(\neg A)$	FA	FA

ν	ν_0
$T\Box A$	TA
$F\Diamond A$	FA

π	π_0
$T\Diamond A$	TA
$F\Box A$	FA

Tableau calculus

Every combination of top-level operator and sign occurs in one of the above cases.

When constructing the tableau, we use signed formulae prefixed by states:

$$\sigma Z A$$

where $Z \in \{T, F\}$, A is a formula, and σ is a finite sequence of natural numbers.

For the modal logic K, σ_1 is accessible from σ iff

$$\sigma_1 = \sigma n \text{ for some natural number } n.$$

Tableau expansion rules are shown on the next slide.

Modal propositional expansion rules

α-**Expansion** (for formulas that are essentially conjunctions: append subformulas α_1 and α_2 one on top of the other)

$$\frac{\sigma\ \alpha}{\begin{array}{c}\sigma\ \alpha_1\\ \sigma\ \alpha_2\end{array}}$$

β-**Expansion** (for formulas that are essentially disjunctions: append β_1 and β_2 horizontally, i. e., branch into β_1 and β_2)

$$\frac{\sigma\ \beta}{\sigma\ \beta_1 \mid \sigma\ \beta_2}$$

Modal propositional expansion rules

ν-**Expansion** (for formulae which are essentially of the form $\sigma\ T\Box A$:
append $\sigma'\nu_0$, such that σ' accessible from σ and occurs on the branch already)

$$\frac{\sigma\ \nu}{\sigma'\ \nu_0}$$

π-**Expansion** (for formulae which are essentially of the form $\sigma\ T\Diamond A$:
append $\sigma'\pi_0$, such that σ' is a simple unrestricted extension of σ, i.e. σ' is accessible from σ and no other prefix of the branch starts with σ')

$$\frac{\sigma\ \pi}{\sigma'\ \pi_0}$$

Tableau calculus

A tableau is closed if every branch contains some pair of formulas of the form $s\ TA$ and $s\ FA$.

A proof for modal logic formula consists of a closed tableau starting with the root $1\ FA$.

Example

These tableau rules can be used to analyze whether $\Box A \to \Diamond A$ is a theorem of K as follows:

1	$F \Box A \to \Diamond A$	(1)
1	$T \Box A$	(2) from 1
1	$F \Diamond A$	(3) from 1

No other proof rules can be used because the modal formulas are ν rules, which are only applicable for accessible prefixes that already occur on the branch.

Example

These tableau rules can be used to analyze whether $\Box A \to \Diamond A$ is a theorem of K as follows:

1	$F \Box A \to \Diamond A$	(1)
1	$T \Box A$	(2) from 1
1	$F \Diamond A$	(3) from 1

No other proof rules can be used because the modal formulas are ν rules, which are only applicable for accessible prefixes that already occur on the branch.

Intuition

The labels denote possible worlds. We can construct a Kripke model $\mathcal{K}$ with one possible world only and the empty relation.

Then $\Box A$ is true in $\mathcal{K}$, but $\Diamond A$ is false, so $\Box A \to \Diamond A$ is false in $\mathcal{K}$.

Example

Without the restriction that the prefix should already appear on the path, we could have closed the tableau as follows:

1	$F\Box A \to \Diamond A$	(1)
1	$T\Box A$	(2) from 1
1	$F\Diamond A$	(3) from 1
11	TA	(4) from 2
11	FA	(5) from 3

But this would have been wrong, since $\Box A \to \Diamond A$ is not a theorem of K.

Tableau calculus

The rules above are sound and complete for the modal logic K.

For other logics it may be necessary to change

- accessibility relation on prefixes

- the two modal rules.

Tableau calculus

The rules above are sound and complete for the modal logic K.

For other logics it may be necessary to change

- accessibility relation on prefixes

- the two modal rules.

A tableau formed using the rules presented before is called a K-tableau.

Example

Prove that $\Box A \wedge \Box B \to \Box(A \wedge B)$

1	$F(\Box A \wedge \Box B) \to \Box(A \wedge B)$	(1)
1	$T\Box A \wedge \Box B$	(2), α, 1_1
1	$F\Box(A \wedge B)$	(3), α, 1_2
1	$T\Box A$	(4), α, 2_1
1	$T\Box B$	(5), α, 2_1
11	$F(A \wedge B)$	(6), π, from 3

11	FA	(7), β, 6_1	11	FB	(8), β, 6_2
11	TA	(9), ν, from 4	11	TB	(10) ν, from 5
	$\perp$	7 and 9		$\perp$	10 and 8

Soundness and Completeness

Definition. A tableau is satisfiable in K if it has a path P, for which there is a Kripke structure $K = (S, R, I)$ for the modal logic K and a mapping m from prefixes of P to S such that

1. $m(s)Rm(s')$ iff prefix s' is accessible from prefix s; and
2. $(K, m(s)) \models A$ for every formula sTA on path P.
3. $(K, m(s)) \models \neg A$ for every formula sFA on path P.

In the sequel we will just abbreviate the last two cases to: $(K, m(s)) \models A$ for every (signed) formula sA on path P.

Soundness and Completeness

Soundness

If FA is satisfiable then we cannot derive $\perp$ on all branches

If we can construct a closed tableau with root FA,
then there is no Kripke structure in which A evaluates to false.

Soundness and Completeness

Soundness

If FA is satisfiable then we cannot derive $\perp$ on all branches

If we can construct a closed tableau with root FA,
then there is no Kripke structure in which A evaluates to false.

Theorem. If there is a closed K-tableau with root $1FA$, then A is valid in all Kripke structures of K.

Soundness and Completeness

Soundness

If FA is satisfiable then we cannot derive $\perp$ on all branches

If we can construct a closed tableau with root FA,
then there is no Kripke structure in which A evaluates to false.

Theorem. If there is a closed K-tableau with root $1FA$, then A is valid in all Kripke structures of K.

In order to prove the theorem we will first prove the following lemma

Lemma. Let T_0 is a K-satisfiable tableau, and let T be the extension of T_0 with one of the extension rules. Then T is a K-satisfiable tableau as well.

Soundness and Completeness

Lemma. Let T_0 is a K-satisfiable tableau, and let T be the extension of T_0 with one of the extension rules. Then T is a K-satisfiable tableau as well.

Proof. We only consider the ν and π rules.

T_0 is satisfiable in K if it has a path P, for which there is a Kripke structure $K = (S, R, I)$ for the modal logic K and a mapping m from prefixes of P to S such that

1. $m(\sigma)Rm(\sigma')$ if prefix σ' is accessible from prefix σ; and
2. $(K, m(\sigma)) \models A$ for every formula sTA on path P.
3. $(K, m(\sigma)) \models \neg A$ for every formula sFA on path P.

Assume first that formula $\sigma\nu$ occurs on path P and the path is extended by the ν rule to $P \cup \{\sigma'\nu_0\}$, where σ' occurs already in P and is accessible from σ.
Then $m(\sigma)Rm(\sigma')$ and $(\mathcal{K}, m(\sigma)) \models \nu$.

From this it immediately follows that $(\mathcal{K}, m(\sigma')) \models \nu_0$.

Soundness and Completeness

Lemma. Let T_0 is a K-satisfiable tableau, and let T be the extension of T_0 with one of the extension rules. Then T is a K-satisfiable tableau as well.

Proof. (continued)
Assume now that formula $\sigma\pi$ occurs on path P and the path is extended by the π rule to $P \cup \{\sigma'\pi_0\}$, where no other prefix of P starts with σ' and σ' is accessible from σ. Then $m(\sigma)Rm(\sigma')$ and $(\mathcal{K}, m(\sigma)) \models \pi$.

From this it immediately follows that there exists $s \in S$ such that $(\mathcal{K}, s) \models \pi_0$.

We extend the map m by defining $m(\sigma') = s$.

(1) By the conditions on the π-rule, we know that σ' is accessible from a prefix ρ on the path P iff $\rho = \sigma$.

(2) Moreover, for every prefix ρ on the path P, ρ is not accessible from σ'.

These properties ensure that for every two prefixes on the path $P \cup \{\sigma'\pi_0\}$ we have:

$m(\rho_1)Rm(\rho_2)$ if ρ_2 is accessible from ρ_1. Thus, T is K-satisfiable.

Soundness and Completeness

Lemma. Let T_0 is a K-satisfiable tableau, and let T be the extension of T_0 with one of the extension rules. Then T is a K-satisfiable tableau as well.

Theorem. If there is a closed K-tableau with root $1FA$, then A is valid in all Kripke structures of K.

Proof. Let T be the closed K-Tableau with root $1FA$. Assume there exists a Kripke-Structure $\mathcal{K} = (S, R, I)$ and $s \in S$ such that $(\mathcal{K}, s) \models \neg A$.

Then the root of T, $1FA$, is a K-satisfiable tableau if we define $m(1) = s$. By the previous Lemma the extension of a K-satisfiable tableau with one of the extension rules is a K-satisfiable tableau as well.

It then follows that T is K-satisfiable, which contradicts the fact that T is closed.

Soundness and Completeness

Completeness

Weak form:

Show that if A is valid then there exists a closed tableau with root $1FA$.

Soundness and Completeness

Completeness

Weak form:

Show that if A is valid then there exists a closed tableau with root $1FA$.

Stronger form:

Would like to show that if $N \models A$ then, if we consider the formulae in N as "axioms" and assume that FA then we can construct a closed tableau.

Soundness and Completeness

Completeness (weak form)

Theorem. If A is valid then there exists a closed tableau with root $1FA$.

Proof. (Idea)

We prove the contrapositive. Assume that every tableau for $1FA$ has an open saturated path P.

Let P_0 the set of all signed formulae with prefixes occurring on P.

Then for every ν-formula $\sigma\nu$, the path contains also the consequence of the ν-rule, $\sigma'\nu_0$, where σ' occurs in P and is accessible from σ.

We construct a Kripke model $\mathcal{K} = (S, R, I)$ for P as follows:

- S is the set of all prefixes occurring on P;
- R is the accessibility relation on the set of prefixes;
- If A propositional variable: $I(A, \sigma) = 1$ iff σTA occurs on P.

Soundness and Completeness

Completeness (weak form)

Theorem. If A is valid then there exists a closed tableau with root $1FA$.

Proof. (Continued)

One can prove by induction on the structure of the signed formulae that for every formula σC on P, $(\mathcal{K}, \sigma) \models C$.

Soundness and Completeness

Completeness (weak form)

Theorem. If A is valid then there exists a closed tableau with root $1FA$.

Proof. (Continued)

One can prove by induction on the structure of the signed formulae that for every formula σC on P, $(\mathcal{K}, \sigma) \models C$.

Example 1:

If $\sigma_0 T\Box B$ occurs in P, then for every prefix $\sigma \in S$ which is reachable from σ_0 also σTB occurs in P.

Induction hypothesis: $(\mathcal{K}, \sigma) \models B$ (and this holds for all $\sigma \in S$ with $\sigma_0 R \sigma$. Thus, $(\mathcal{K}, \sigma_0) \models \Box B$.

Soundness and Completeness

Completeness (weak form)

Theorem. If A is valid then there exists a closed tableau with root $1FA$.

Proof. (Continued)

One can prove by induction on the structure of the signed formulae that for every formula σC on P, $(\mathcal{K}, \sigma) \models C$.

Example 2:

If $\sigma_0 F \Box B$ occurs in P, there exists a prefix σ accessible from σ_0 such that σFB occurs in P.

By induction hypothesis, $(\mathcal{K}, \sigma) \models FB$ (i.e. $(\mathcal{K}, \sigma) \models \neg B$, hence $(\mathcal{K}, \sigma_0) \models F\Box B$.

Completeness

Completeness (strong form)

Would like to show that if $N \models A$ then, if we consider the formulae in N as "axioms" and assume that FA then we can construct a closed tableau.

We defined "local entailment" and "global entailment"

$\mapsto$ We distinguish L-completeness and G-completeness

Entailment

Global entailment:

$N \models_G F$ iff for every Kripke structure $\mathcal{K} = (S, R, I)$:

$$\text{If } \mathcal{K} \models G \text{ for every } G \in N \text{ then } \mathcal{K} \models F$$

Local entailment:

$N \models_L F$ iff for every Kripke structure $\mathcal{K} = (S, R, I)$ and every $s \in S$:

$$\text{If } (\mathcal{K}, s) \models G \text{ for every } G \in N \text{ then } (\mathcal{K}, s) \models F$$

L-Completeness

Let N be a set of modal formulae.

Definition A K-tableau is an K-L-Tableau over N if for every formula $B \in N$ the following rule can be used:

$$\frac{}{1TB}$$

Theorem. Let N be a set of modal formulae and A a modal logic formula. Then $N \models_L A$ iff there exists a closed K-L-Tableau with root $1FA$.

G-Completeness

Let N be a set of modal formulae.

Definition A K-tableau is an K-G-Tableau over N if for every formula $B \in N$ and for every prefix σ on the current path the following rule can be used:

$$\frac{}{\sigma TB}$$

Theorem. Let N be a set of modal formulae and A a modal logic formula. Then $N \models_G A$ iff there exists a closed K-G-Tableau with root $1FA$.

Tableau calculi

Sound and complete tableau calculi can be devised for a large class of systems of propositional modal logic.

Main challenge: Prove termination (can construct "saturated" or closed model in a finite number of steps)

"Blocking techniques"

Theorem proving in modal logics

- Inference system (soundness and completeness results)
- Tableau calculi (soundness and completeness results)
- Translation to first order logic (+ e.g. Resolution)

Translation for classical logic

$\mathcal{K} = (S, R, I)$ Kripke model

$$
\begin{array}{rcll}
\mathrm{val}_{\mathcal{K}}(\bot)(s) & = & 0 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(\top)(s) & = & 1 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(P)(s) = 1 & \leftrightarrow & I(P)(s) = 1 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(\neg F)(s) = 1 & \leftrightarrow & \mathrm{val}_{\mathcal{K}}(F)(s) = 0 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(F_1 \wedge F_2)(s) = 1 & \leftrightarrow & \mathrm{val}_{\mathcal{K}}(F_1)(s) \wedge \mathrm{val}_{\mathcal{K}}(F_1)(s) = 1 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(F_1 \vee F_2)(s) = 1 & \leftrightarrow & \mathrm{val}_{\mathcal{K}}(F_1)(s) \vee \mathrm{val}_{\mathcal{K}}(F_1)(s) = 1 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(\Box F)(s) = 1 & \leftrightarrow & \forall s'(R(s, s') \rightarrow \mathrm{val}_{\mathcal{K}}(F)(s') = 1 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(\Diamond F)(s) = 1 & \leftrightarrow & \exists s'(R(s, s') \text{ and } \mathrm{val}_{\mathcal{K}}(F)(s') = 1 & \text{for all } s
\end{array}
$$

Translation for classical logic

$\mathcal{K} = (S, R, I)$ Kripke model

$$
\begin{array}{rcll}
\mathsf{val}_{\mathcal{K}}(\bot)(s) & = & 0 & \text{for all } s \\
\mathsf{val}_{\mathcal{K}}(\top)(s) & = & 1 & \text{for all } s \\
\mathsf{val}_{\mathcal{K}}(P)(s) = 1 & \leftrightarrow & I(P)(s) = 1 & \text{for all } s \\
\mathsf{val}_{\mathcal{K}}(\neg F)(s) = 1 & \leftrightarrow & \mathsf{val}_{\mathcal{K}}(F)(s) = 0 & \text{for all } s \\
\mathsf{val}_{\mathcal{K}}(F_1 \wedge F_2)(s) = 1 & \leftrightarrow & \mathsf{val}_{\mathcal{K}}(F_1)(s) \wedge \mathsf{val}_{\mathcal{K}}(F_1)(s) = 1 & \text{for all } s \\
\mathsf{val}_{\mathcal{K}}(F_1 \vee F_2)(s) = 1 & \leftrightarrow & \mathsf{val}_{\mathcal{K}}(F_1)(s) \vee \mathsf{val}_{\mathcal{K}}(F_1)(s) = 1 & \text{for all } s \\
\mathsf{val}_{\mathcal{K}}(\Box F)(s) = 1 & \leftrightarrow & \forall s'(R(s, s') \rightarrow \mathsf{val}_{\mathcal{K}}(F)(s') = 1 & \text{for all } s \\
\mathsf{val}_{\mathcal{K}}(\Diamond F)(s) = 1 & \leftrightarrow & \exists s'(R(s, s') \text{ and } \mathsf{val}_{\mathcal{K}}(F)(s') = 1 & \text{for all } s
\end{array}
$$

$$
\begin{array}{llcl}
\mathit{Translation}: & P \in \Pi & \mapsto & P/1 \text{ unary predicate} \\
 & F \text{ formula} & \mapsto & P_F/1 \text{ unary predicate} \\
 & R \text{ acc.rel} & \mapsto & R/2 \text{ binary predicate}
\end{array}
$$

Translation for classical logic

$\mathcal{K} = (S, R, I)$ Kripke model

$$
\begin{array}{rcll}
\mathrm{val}_{\mathcal{K}}(\bot)(s) & = & 0 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(\top)(s) & = & 1 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(P)(s) = 1 & \leftrightarrow & I(P)(s) = 1 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(\neg F)(s) = 1 & \leftrightarrow & \mathrm{val}_{\mathcal{K}}(F)(s) = 0 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(F_1 \wedge F_2)(s) = 1 & \leftrightarrow & \mathrm{val}_{\mathcal{K}}(F_1)(s) \wedge \mathrm{val}_{\mathcal{K}}(F_1)(s) = 1 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(F_1 \vee F_2)(s) = 1 & \leftrightarrow & \mathrm{val}_{\mathcal{K}}(F_1)(s) \vee \mathrm{val}_{\mathcal{K}}(F_1)(s) = 1 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(\Box F)(s) = 1 & \leftrightarrow & \forall s'(R(s, s') \rightarrow \mathrm{val}_{\mathcal{K}}(F)(s') = 1 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(\Diamond F)(s) = 1 & \leftrightarrow & \exists s'(R(s, s') \text{ and } \mathrm{val}_{\mathcal{K}}(F)(s') = 1 & \text{for all } s
\end{array}
$$

Translation:

$$
\begin{array}{lcl}
P \in \Pi & \mapsto & P/1 \text{ unary predicate} \\
F \text{ formula} & \mapsto & P_F/1 \text{ unary predicate} \\
R \text{ acc.rel} & \mapsto & R/2 \text{ binary predicate} \\
\mathrm{val}_{\mathcal{K}}(P)(s) = 1 & \mapsto & P(s) \\
\mathrm{val}_{\mathcal{K}}(P)(s) = 0 & \mapsto & \neg P(s)
\end{array}
$$

$$
\begin{array}{rcl}
\forall s(P_{\neg F}(s) & \leftrightarrow & \neg P_F(s)) \\
\forall s(P_{F_1 \wedge F_2}(s) & \leftrightarrow & P_{F_1}(s) \wedge P_{F_2}(s)) \\
\forall s(P_{F_1 \vee F_2}(s) & \leftrightarrow & P_{F_1}(s) \vee P_{F_2}(s)) \\
\forall s(P_{\Box F}(s) & \leftrightarrow & \forall s'(R(s, s') \rightarrow P_F(s'))) \\
\forall s(P_{\Diamond F}(s) & \leftrightarrow & \exists s'(R(s, s') \wedge P_F(s')))
\end{array}
$$

Translation for classical logic

$\mathcal{K} = (S, R, I)$ Kripke model

$$
\begin{array}{rcll}
\mathrm{val}_{\mathcal{K}}(\bot)(s) & = & 0 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(\top)(s) & = & 1 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(P)(s) = 1 & \leftrightarrow & I(P)(s) = 1 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(\neg F)(s) = 1 & \leftrightarrow & \mathrm{val}_{\mathcal{K}}(F)(s) = 0 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(F_1 \wedge F_2)(s) = 1 & \leftrightarrow & \mathrm{val}_{\mathcal{K}}(F_1)(s) \wedge \mathrm{val}_{\mathcal{K}}(F_1)(s) = 1 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(F_1 \vee F_2)(s) = 1 & \leftrightarrow & \mathrm{val}_{\mathcal{K}}(F_1)(s) \vee \mathrm{val}_{\mathcal{K}}(F_1)(s) = 1 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(\Box F)(s) = 1 & \leftrightarrow & \forall s'(R(s, s') \rightarrow \mathrm{val}_{\mathcal{K}}(F)(s') = 1 & \text{for all } s \\
\mathrm{val}_{\mathcal{K}}(\Diamond F)(s) = 1 & \leftrightarrow & \exists s'(R(s, s') \text{ and } \mathrm{val}_{\mathcal{K}}(F)(s') = 1 & \text{for all } s
\end{array}
$$

Translation: Given F modal formula:

$$
\begin{array}{lcl}
P \in \Pi & \mapsto & P/1 \text{ unary predicate} \\
F' \text{ subformula of F} & \mapsto & P_F/1 \text{ unary predicate} \\
R \text{ acc.rel} & \mapsto & R/2 \text{ binary predicate} \\
\mathrm{val}_{\mathcal{K}}(P)(s) = 1 & \mapsto & P(s) \\
\mathrm{val}_{\mathcal{K}}(P)(s) = 0 & \mapsto & \neg P(s)
\end{array}
$$

$$
\begin{array}{rcl}
\forall s(P_{\neg F'}(s) & \leftrightarrow & \neg P_{F'}(s)) \\
\forall s(P_{F_1 \wedge F_2}(s) & \leftrightarrow & P_{F_1}(s) \wedge P_{F_2}(s)) \\
\forall s(P_{F_1 \vee F_2}(s) & \leftrightarrow & P_{F_1}(s) \vee P_{F_2}(s)) \\
\forall s(P_{\Box F'}(s) & \leftrightarrow & \forall s'(R(s, s') \rightarrow P_{F'}(s'))) \\
\forall s(P_{\Diamond F'}(s) & \leftrightarrow & \exists s'(R(s, s') \wedge P_{F'}(s')))
\end{array}
$$

where the index formulae range over all subfromulae of F.

Translation to classical logic

Translation: Given F modal formula:

$$
\begin{array}{lcl}
P \in \Pi & \mapsto & P/1 \text{ unary predicate} \\
F' \text{ subformula of F} & \mapsto & P_{F'}/1 \text{ unary predicate} \\
R \text{ acc.rel} & \mapsto & R/2 \text{ binary predicate} \\
\mathrm{val}_{\mathcal{K}}(P)(s) = 1 & \mapsto & P(s) \\
\mathrm{val}_{\mathcal{K}}(P)(s) = 0 & \mapsto & \neg P(s)
\end{array}
$$

$$
\underbrace{
\begin{array}{rcl}
\forall s(P_{\neg F'}(s) & \leftrightarrow & \neg P_{F'}(s)) \\
\forall s(P_{F_1 \wedge F_2}(s) & \leftrightarrow & P_{F_1}(s) \wedge P_{F_2}(s)) \\
\forall s(P_{F_1 \vee F_2}(s) & \leftrightarrow & P_{F_1}(s) \vee P_{F_2}(s)) \\
\forall s(P_{\Box F'}(s) & \leftrightarrow & \forall s'(R(s, s') \rightarrow P_{F'}(s'))) \\
\forall s(P_{\Diamond F'}(s) & \leftrightarrow & \exists s'(R(s, s') \wedge P_{F'}(s'))) \\
\multicolumn{3}{l}{\text{where the index formulae range over all subformulae of } F.}
\end{array}
}_{\mathrm{Rename}(F)}
$$

Theorem.
F is K-satisfiable iff $\exists x P_F(x) \wedge \mathrm{Rename}(F)$ is satisfiable in first-order logic.

Translation to classical logic

Example

To prove that $F := \Box(P \wedge Q) \rightarrow \Box P \wedge \Box Q$ is K-valid

The following are equivalent:

(1) F is valid

(2) $\neg F := \Box(P \wedge Q) \wedge \neg(\Box P \wedge \Box Q))$ is unsatisfiable

(3) $\exists x P_{\neg F}(x) \wedge \text{Rename}(\neg F)$ is unsatisfiable

Translation to classical logic

Example

The following are equivalent:
(2) $\neg F := \Box(P \wedge Q) \wedge \neg(\Box P \wedge \Box Q))$ is unsatisfiable
(3) $\exists x P_{\neg F}(x) \wedge \text{Rename}(\neg F)$ is unsatisfiable

$$
\begin{array}{ll}
\exists x & P_{\Box(P\wedge Q)\wedge\neg(\Box P\wedge\Box Q))}(x) \\
\forall x & (P_{\Box(P\wedge Q)\wedge\neg(\Box P\wedge\Box Q)}(x) \leftrightarrow P_{\Box(P\wedge Q)}(x) \wedge P_{\neg(\Box P\wedge\Box Q)}(x)) \\
\forall x & (P_{\neg(\Box P\wedge\Box Q)}(x) \leftrightarrow \neg P_{\Box P\wedge\Box Q}(x)) \\
\forall x & (P_{\Box P\wedge\Box Q}(x) \leftrightarrow P_{\Box P}(x) \wedge P_{\Box Q}(x)) \\
\forall x & (P_{\Box P}(x) \leftrightarrow \forall y(R(x,y) \rightarrow P(y))) \\
\forall x & (P_{\Box Q}(x) \leftrightarrow \forall y(R(x,y) \rightarrow Q(y))) \\
\forall x & (P_{\Box(P\wedge Q)}(x) \leftrightarrow \forall y(R(x,y) \rightarrow P_{P\wedge Q}(y))) \\
\forall x & (P_{P\wedge Q}(x) \leftrightarrow P(x) \wedge Q(x))
\end{array}
$$

Translation to classical logic

Example

The following are equivalent:
(2) $\neg F := \Box(P \wedge Q) \wedge \neg(\Box P \wedge \Box Q))$ is unsatisfiable
(3) $\exists x P_{\neg F}(x) \wedge \text{Rename}(\neg F)$ is unsatisfiable

Prenex normal form

$\exists x$	$P_{\Box(P\wedge Q)\wedge\neg(\Box P\wedge\Box Q))}(x)$
$\forall x$	$(P_{\Box(P\wedge Q)\wedge\neg(\Box P\wedge\Box Q)}(x) \leftrightarrow P_{\Box(P\wedge Q)}(x) \wedge P_{\neg(\Box P\wedge\Box Q)}(x))$
$\forall x$	$(P_{\neg(\Box P\wedge\Box Q)}(x) \leftrightarrow \neg P_{\Box P\wedge\Box Q}(x))$
$\forall x$	$(P_{\Box P\wedge\Box Q}(x) \leftrightarrow P_{\Box P}(x) \wedge P_{\Box Q}(x))$
$\forall x \forall y$	$(P_{\Box P}(x) \rightarrow (R(x,y) \rightarrow P(y)))$
$\forall x \exists y$	$(R(x,y) \rightarrow P(y)) \rightarrow P_{\Box P}(x))$
$\forall x \forall y$	$(P_{\Box Q}(x) \rightarrow (R(x,y) \rightarrow Q(y)))$
$\forall x \exists y$	$(R(x,y) \rightarrow Q(y)) \rightarrow P_{\Box Q}(x))$
$\forall x \forall y$	$(P_{\Box(P\wedge Q)}(x) \rightarrow (R(x,y) \rightarrow P_{P\wedge Q}(y)))$
$\forall x \exists y$	$(R(x,y) \rightarrow P_{P\wedge Q}(y)) \rightarrow P_{\Box(P\wedge Q)}(x)$
$\forall x$	$(P_{P\wedge Q}(x) \leftrightarrow P(x) \wedge Q(x))$

Translation to classical logic

Example

The following are equivalent:
(2) $\neg F := \Box(P \wedge Q) \wedge \neg(\Box P \wedge \Box Q))$ is unsatisfiable
(3) $\exists x P_{\neg F}(x) \wedge \text{Rename}(\neg F)$ is unsatisfiable

Skolemization

$$
\begin{array}{ll}
 & P_{\Box(P \wedge Q) \wedge \neg(\Box P \wedge \Box Q))}(c) \\
\forall x & (P_{\Box(P \wedge Q) \wedge \neg(\Box P \wedge \Box Q)}(x) \leftrightarrow P_{\Box(P \wedge Q)}(x) \wedge P_{\neg(\Box P \wedge \Box Q)}(x)) \\
\forall x & (P_{\neg(\Box P \wedge \Box Q)}(x) \leftrightarrow \neg P_{\Box P \wedge \Box Q}(x)) \\
\forall x & (P_{\Box P \wedge \Box Q}(x) \leftrightarrow P_{\Box P}(x) \wedge P_{\Box Q}(x)) \\
\forall x \forall y & (P_{\Box P}(x) \rightarrow (R(x, y) \rightarrow P(y))) \\
\forall x & (R(x, f_1(x) \rightarrow P(f_1(x))) \rightarrow P_{\Box P}(x)) \\
\forall x \forall y & (P_{\Box Q}(x) \rightarrow (R(x, y) \rightarrow Q(y))) \\
\forall x & (R(x, f_2(x)) \rightarrow Q(f_2(x))) \rightarrow P_{\Box Q}(x)) \\
\forall x \forall y & (P_{\Box(P \wedge Q)}(x) \rightarrow (R(x, y) \rightarrow P_{P \wedge Q}(y))) \\
\forall x & (R(x, f_3(x)) \rightarrow P_{P \wedge Q}(f_3(x))) \rightarrow P_{\Box(P \wedge Q)}(x) \\
\forall x & (P_{P \wedge Q}(x) \leftrightarrow P(x) \wedge Q(x))
\end{array}
$$

CNF translation, Resolution Exploit polarity!!!

Another example

Task: Check if there exists a Kripke model such that $F = \Diamond(Q \to \Diamond Q)$ holds at some state in this Kripke model.

$P_F(c)$

$\forall x(P_F(x) \leftrightarrow \exists y(R(x,y) \land P_{Q \to \Diamond Q}(y)))$

$\forall x(P_{Q \to \Diamond Q}(x) \leftrightarrow (Q(x) \to P_{\Diamond Q}(x)))$

$\forall x(P_{\Diamond Q}(x) \leftrightarrow \exists y(R(x,y) \land Q(y)))$

Another example

Task: Check if there exists a Kripke model such that $F = \Diamond(Q \rightarrow \Diamond Q)$ holds at some state in this Kripke model.

P_F, $P_{Q\rightarrow\Diamond Q}$, $P_{\Diamond Q}$: positive polarity!

$P_F(c)$

$\forall x(P_F(x) \rightarrow \exists y(R(x,y) \wedge P_{Q\rightarrow\Diamond Q}(y)))$

$\forall x((P_{Q\rightarrow\Diamond Q}(x) \rightarrow (Q(x) \rightarrow P_{\Diamond Q}(x)))$

$\forall x(P_{\Diamond Q}(x) \rightarrow \exists y(R(x,y) \wedge Q(y)))$

Another example

Task: Check if there exists a Kripke model such that $F = \Diamond(Q \rightarrow \Diamond Q)$ holds at some state in this Kripke model.

Prenex, Skolemization

$P_F(c)$

$\forall x(P_F(x) \rightarrow (R(x, f(x)) \land P_{Q \rightarrow \Diamond Q}(f(x))))$

$\forall x(P_{Q \rightarrow \Diamond Q}(x) \rightarrow (Q(x) \rightarrow P_{\Diamond Q}(x))$

$\forall x(P_{\Diamond Q} \rightarrow (R(x, g(x)) \land Q(g(x))))$

Another example

Task: Check if there exists a Kripke model such that $F = \Diamond(Q \rightarrow \Diamond Q)$ holds at some state in this Kripke model.

CNF

$P_F(c)$

$\neg P_F(x) \vee R(x, f(x))$

$\neg P_F(x) \vee P_{Q\rightarrow\Diamond Q}(f(x)))$

$\neg P_{Q\rightarrow\Diamond Q}(x) \vee \neg Q(x) \vee P_{\Diamond Q}(x)$

$\neg P_{\Diamond Q}(x) \vee R(x, g(x))$

$\neg P_{\Diamond Q}(x) \vee Q(g(x))))$

Resolution

Resolution for General Clauses

General binary resolution ***Res*****:**

$$\frac{C \vee A \qquad D \vee \neg B}{(C \vee D)\sigma} \quad \text{if } \sigma = \text{mgu}(A, B) \qquad \text{[resolution]}$$

$$\frac{C \vee A \vee B}{(C \vee A)\sigma} \quad \text{if } \sigma = \text{mgu}(A, B) \quad \text{[factorization]}$$

Resolution for General Clauses

For inferences with more than one premise, we assume that the variables in the premises are (bijectively) renamed such that they become different to any variable in the other premises.

We do not formalize this. Which names one uses for variables is otherwise irrelevant.

Ordered resolution with selection

A selection function is a mapping

$$S : C \quad \mapsto \quad \text{set of occurrences of } \textit{negative} \text{ literals in } C$$

Example of selection with selected literals indicated as $\boxed{X}$:

$$\boxed{\neg A} \vee \neg A \vee B \qquad\qquad \boxed{\neg B_0} \vee \boxed{\neg B_1} \vee A$$

Let $\succ$ be a total and well-founded ordering on ground atoms. Then $\succ$ can be extended to a total and well-founded ordering on ground literals and clauses

A literal L (possibly with variables) is called [strictly] maximal in a clause C if and only if there exists a ground substitution σ such that for all L' in C: $L\sigma \succeq L'\sigma$ [$L\sigma \succ L'\sigma$].

Resolution Calculus $Res_S^{\succ}$

Let $\succ$ be an atom ordering and S a selection function.

$$\frac{C \lor A \qquad \neg B \lor D}{(C \lor D)\sigma} \qquad \text{[ordered resolution with selection]}$$

if $\sigma = \mathrm{mgu}(A, B)$ and

(i) $A\sigma$ strictly maximal wrt. $C\sigma$;

(ii) nothing is selected in C by S;

(iii) either $\neg B$ is selected,
or else nothing is selected in $\neg B \lor D$ and $\neg B\sigma$ is maximal in $D\sigma$.

Resolution Calculus $Res_S^{\succ}$

$$\frac{C \vee A \vee B}{(C \vee A)\sigma} \qquad \text{[ordered factoring]}$$

if $\sigma = \mathrm{mgu}(A, B)$ and $A\sigma$ is maximal in $C\sigma$ and nothing is selected in C.

Soundness and Refutational Completeness

Theorem:

Let $\succ$ be an atom ordering and S a selection function such that $Res_S^{\succ}(N) \subseteq N$. Then

$$N \models \bot \Leftrightarrow \bot \in N$$

Ordered resolution for modal logics

It has been proved that ordered resolution (possibly with selection) can be used as a decision procedure for the propositional modal logic K and also for many extensions of K.

Goal: Define ordering/selection function such that few inferences can take place, and such that the size of terms/length of clauses cannot grow in the resolvents.

Decidability of modal logics

Decidability of modal logics

- **Direct approach:** Prove finite model property

  If a formula F is satisfiable then it has a model with at least $f(\mathit{size}(F))$ elements, where f is a concrete function.

  Generate all models with $1, 2, 3, \ldots, f(\mathit{size}(F))$ elements.

Decidability of modal logics

- **Direct approach:** Prove finite model property

  If a formula F is satisfiable then it has a model with at least $f(\mathit{size}(F))$ elements, where f is a concrete function.

  Generate all models with $1, 2, 3, \dots, f(\mathit{size}(F))$ elements.

- **Alternative approaches:**
  - Show that terminating sound and complete tableau calculi exist
  - Show that ordered resolution ($+$ additional refinements) terminates on the type of first-order formulae which are generated starting from a modal formula.

Decidability

Direct approach

Idea:

We show that if a formula A has n subformulae, then
$\vdash_K A$ iff, A is valid in all frames having at most 2^n elements.

or alternatively, that the following are equivalent:

(1) There exists a Kripke structure $\mathcal{K} = (S, R, I)$ and $s \in S$ such that $(\mathcal{K}, s) \models A$.

(2) There exists a Kripke structure $\mathcal{K}' = (S', R', I')$ and $s' \in S'$ s.t.:
- $(\mathcal{K}', s') \models A$
- S' consists of at most 2^n states.

Decidability

Idea:

We show that if a formula A has n subformulae, then
$\vdash_K A$ iff A is valid in all frames having at most 2^n elements.

or alternatively, that the following are equivalent:

(1) There exists a Kripke structure $\mathcal{K} = (S, R, I)$ and $s \in S$ such that $(\mathcal{K}, s) \models A$.

(2) There exists a Kripke structure $\mathcal{K}' = (S', R', I')$ and $s' \in S'$ such that:
- $(\mathcal{K}', s') \models A$
- S' consists of at most 2^n states.

Goal: Construct the finite Kripke structure $\mathcal{K}'$ starting from $\mathcal{K}$.

Decidability

Filtrations

Fix a model $\mathcal{K} = (S, R, I)$ and a set $\Gamma \subseteq Fma_\Sigma$ that is closed under subformulae, i.e. $B \in \Gamma$ implies Subformulae$(B) \subseteq \Gamma$.

For each $s \in S$, define

$$\Gamma_s = \{B \in \Gamma \mid (\mathcal{K}, s) \models B\}$$

and put $s \sim_\Gamma t$ iff $\Gamma_s = \Gamma_t$,

Decidability

Filtrations

Fix a model $\mathcal{K} = (S, R, I)$ and a set $\Gamma \subseteq Fma_\Sigma$ that is closed under subformulae, i.e. $B \in \Gamma$ implies Subformulae$(B) \subseteq \Gamma$.

For each $s \in S$, define

$$\Gamma_s = \{B \in \Gamma \mid (\mathcal{K}, s) \models B\}$$

and put $s \sim_\Gamma t$ iff $\Gamma_s = \Gamma_t$,

Then $s \sim_\Gamma t$ iff for all $B \in \Gamma$, $(\mathcal{K}, s) \models B$ iff $(\mathcal{K}, t) \models B$.

Fact: $\sim_\Gamma$ is an equivalence relation on S.

Decidability

Let $[s] = \{t \mid s \sim_\Gamma t\}$ be the $\sim_\Gamma$-equivalence class of s.

Let $S_\Gamma := \{[s] \mid s \in S\}$ be the set of all such equivalence classes.

Lemma. If Γ is finite, then S_Γ is finite and has at most 2^n elements, where n is the number of elements of Γ.

Proof. Let $f : S_\Gamma \to \mathcal{P}(\Gamma)$ be defined by $f([s]){=}\Gamma_s{=}\{B{\in}\Gamma \mid (\mathcal{K}, s) \models B\}$.

Since $[s] = [t]$ iff $s \sim_\Gamma t$ iff $\Gamma_s = \Gamma_t$, f is well-defined and one-to-one.

Hence S_Γ has no more elements than there are subsets of Γ.

But if Γ has n elements, then it has 2^n subsets, so S_Γ has at most 2^n elements.

Decidability

Goal: $(\mathcal{K}, s) \models A \quad \mapsto \quad (\mathcal{K}', s') \models A$, $\mathcal{K}' = (S', R', I')$, $\quad |S'| \leq 2^n$.

Step 1: Determine S':

$S' := S_\Gamma$, where $\Gamma = \text{Subformulae}(S)$

Decidability

Goal: $(\mathcal{K}, s) \models A \quad \mapsto \quad (\mathcal{K}', s') \models A$, $\mathcal{K}' = (S', R', I')$, $\quad |S'| \leq 2^n$.

Step 1: Determine S':
$S' := S_\Gamma$, where $\Gamma = \text{Subformulae}(S)$

Step 2: Determine I':
Let $\Pi' = \Pi \cap \Gamma$ the set of all atomic formulae occurring in Γ.
Define $I' : \Pi' \times S' \to \{0, 1\}$ by $I'(P, [s]) = I(P, s)$

Remark: I' well defined (if $s \sim_\Gamma t$ and $P \in \Gamma$ then $I(P, s) = I(P, t)$).

Decidability

Goal: $(\mathcal{K}, s) \models A \quad \mapsto \quad (\mathcal{K}', s') \models A$, $\mathcal{K}' = (S', R', I')$, $\quad |S'| \leq 2^n$.

Step 1: Determine S':
$S' := S_\Gamma$, where $\Gamma = \text{Subformulae}(S)$

Step 2: Determine I':
Let $\Pi' = \Pi \cap \Gamma$ the set of all atomic formulae occurring in Γ.
Define $I' : \Pi' \times S' \to \{0, 1\}$ by $I'(P, [s]) = I(P, s)$

Remark: I' well defined (if $s \sim_\Gamma t$ and $P \in \Gamma$ then $I(P, s) = I(P, t)$).

Step 3: Determine $R' \subseteq S' \times S'$.
Define e.g. $([s], [t]) \in R'$ iff $\exists s' \in [s], \exists t' \in [t]$: $(s', t') \in R$

Decidability

Goal: $(\mathcal{K}, s) \models A \quad \mapsto \quad (\mathcal{K}', s') \models A$, $\mathcal{K}' = (S', R', I')$, $\quad |S'| \leq 2^n$.

Step 1: $S' := S_\Gamma$, where $\Gamma = \text{Subformulae}(S)$

Step 2: $I' : (\Pi \cap \Gamma) \times S' \to \{0, 1\}$ def. by $I'(P, [s]) = I(P, s)$

Step 3: R' def. e.g. by: $([s], [t]) \in R'$ iff $\exists s' \in [s], \exists t' \in [t]$: $(s', t') \in R$

Remark: R' has the following properties:

(F1) if $(s, t) \in R$ then $([s], [t]) \in R'$

(F2) if $([s], [t]) \in R'$ then for all B, if $\Box B \in \Gamma$ and $(\mathcal{K}, s) \models \Box B$, then $(\mathcal{K}, t) \models B$.

Proof: $(F2)$ Assume $([s], [t]) \in R'$. Then $(s', t') \in R$ for $s' \in [s], t' \in [t]$.
Hence if $(\mathcal{K}, s) \models \Box B$ then $(\mathcal{K}, s') \models \Box B$, so $(\mathcal{K}, t') \models B$, i.e. $(\mathcal{K}, t) \models B$.

Decidability

Goal: $(\mathcal{K}, s) \models A \quad \mapsto \quad (\mathcal{K}', s') \models A, \mathcal{K}' = (S', R', I'), \quad |S'| \leq 2^n.$

Step 1: $S' := S_\Gamma$, where $\Gamma = \text{Subformulae}(S)$

Step 2: $I' = I_\Gamma : (\Pi \cap \Gamma) \times S' \to \{0, 1\}$ def. by $I_\Gamma(P, [s]) = I(P, s)$

Step 3: $R' = \{([s], [t]) \mid \exists s' \sim_\Gamma s, \exists t' \sim_\Gamma t s.t. (s', t') \in R\}$

Remark: R' has the following properties:

(F1) if $(s, t) \in R$ then $([s], [t]) \in R'$

(F2) if $([s], [t]) \in R'$ then for all B, if $\Box B \in \Gamma$ and $(\mathcal{K}, s) \models \Box B$, then $(\mathcal{K}, t) \models B$.

Any Kripke structure $\mathcal{K}' = (S_\Gamma, R', I_\Gamma)$ in which R' satisfies (Fl) and (F2) is called a Γ-filtration of $\mathcal{K}$.

Decidability

Examples of filtrations

- The smallest filtration.
  $([s],[t]) \in R'$ iff $\exists s' \sim_\Gamma s, \exists t' \sim_\Gamma t (s', t') \in R$.
- The largest filtration.
  $([s],[t]) \in R$ iff for all $B, \Box B \in \Gamma$, $(\mathcal{K}, s) \models \Box B$ implies $(\mathcal{K}, t) \models B$.
- The transitive filtration.
  $([s],[t]) \in R'$ iff for all $B, \Box B \in \Gamma$, $(\mathcal{K}, s) \models \Box B$ implies $(\mathcal{K}, t) \models \Box B \wedge B$.

When defining $\mathcal{K}'$ we can choose also the second or third definition of R'.

Decidability

Filtration Lemma.

Let Γ be a set of modal formulae closed under subformulae.
Let $\mathcal{K} = (S, R, I)$ be a Kripke structure and let $\mathcal{K}' = (S_\Gamma, R', I_\Gamma)$ be a Γ-filtration of $\mathcal{K}$.

If $B \in \Gamma$, then for any $s \in S$,

$$(\mathcal{K}, s) \models B \quad \text{iff} \quad (\mathcal{K}', [s]) \models B$$

Proof. The case $B = P \in \Pi \cap \Gamma$ is given by the definition of I'

The inductive case for the connectives $\{\wedge, \vee, \neg\}$ is straightforward.

The inductive case for $\Box$ uses (FI) and (F2).

Note that the closure of Γ under subformulae is needed in order to be able to apply the induction hypothesis.

Decidability

Theorem. Let A be a formula with n subformulae.
Then $\vdash_K A$ iff A is valid in all frames having at most 2^n elements.

Proof. Suppose $\not\vdash_K A$. Then there is a model $\mathcal{K} = (S, R, I)$ and a state $s \in S$ at which A is false. Let $\Gamma = \text{Subformulae}(A)$.

Then Γ is closed under subformulae, so we can construct Γ-filtrations $\mathcal{K}' = (S_\Gamma, R', I_\Gamma)$ as above. By the Filtration Lemma, A is false at $[s]$ in any such model, and hence not valid in the frame (S_Γ, R').

We previously showed that the desired bound on the size of S_Γ is 2^n.

Decidability

A logic $\mathcal{L}$ characterized by a set $\mathcal{F}$ of frames* has the finite frame property if it is determined by its finite frames, i.e.,

if $\nvdash_{\mathcal{L}} A$, then there is a finite frame $F \in \mathcal{F}$ s.t. $\mathcal{F} \not\models A$

We showed that the smallest normal logic K has the finite frame property, and a computable bound was given on the size of the invalidating frame for a given non-theorem.

* We can choose $\mathcal{F}$ to be the class of all frames in which all theorems of $\mathcal{L}$ are valid.

Decidability

This implies that the property of K-theoremhood is decidable, i.e. that there is an algorithm for determining, for each formula A, whether or not $\vdash_K A$:

If A has n subformulae, we simply check to see whether or not A is valid in all frames of size at most 2^n.

- Since a finite set has finitely many binary relations (2^{m^2} relations on an m-element set), there are only finitely many frames of size at most 2^n.
- Moreover, to determine whether A is valid on a finite frame F, we need only look at models $I : \Pi \cap \mathrm{Subformulae}(A) \to \{0, 1\}$ on F.

But there are only finitely many such models on F. Thus the whole checking procedure for validity of A in frames of size at most 2^n can be completed in a finite amount of time.