reducing (modal) sequents: propositional part

\( p, q_1, \ldots, q_n \Rightarrow p, r_1, \ldots, r_m \) is valid

\( A, \neg \phi \Rightarrow B \) if and only if \( A \Rightarrow \phi, B \)

\( A \Rightarrow \neg \phi, B \) if and only if \( A, \phi \Rightarrow B \)

\( A, \phi \land \psi \Rightarrow B \) if and only if \( A, \phi, \psi \Rightarrow B \)

\( A \Rightarrow \phi \land \psi, B \) if and only if both \( A \Rightarrow \phi, B \) and \( A \Rightarrow \psi, B \)

\( A, \phi \lor \psi \Rightarrow B \) if and only if both \( A, \phi \Rightarrow B \) and \( A, \psi \Rightarrow B \)

\( A \Rightarrow \phi \lor \psi, B \) if and only if \( A \Rightarrow \phi, \psi, B \)

gives a decision procedure for propositional logic: \( \phi \) valid iff \( \Rightarrow \phi \) valid
intuition and notation

we work in sequent calculus
we start with the intended conclusion ⊢ φ
we try to build a proof while moving upwards

notation: make clear what is the formula,
give every step of the sequent transformation,
indicate clearly your conclusion;
you may rewrite the formula also inside the sequent transformation

example

φ = (p ∧ q) → (p ∨ q) ≡ ¬(p ∧ q) ∨ (p ∨ q)
⇒ ¬(p ∧ q) ∨ (p ∨ q)
⇒ ¬(p ∧ q), (p ∨ q)
p ∧ q ⇒ p ∨ q
p, q ⇒ p ∨ q
p, q ⇒ p, q

the sequent is valid so φ is valid

transformation of sequents: intuition

p₁, . . . , pₙ, ◊φ₁, . . . , ◊φₘ ⇒ q₁, . . . , qₖ, ◊ψ₁, . . . , ◊ψₖ

either we can decide validity because of the propositional part
if not, we go to a next world, leaving behind the propositional part
we then see if φᵢ ⇒ ψ₁, . . . , ψₖ is valid for some i ∈ {1, . . . , m}

the sequent is not valid so φ is not valid
we get a sequent of the form
\[ p_1, \ldots, p_n, \Diamond \phi_1, \ldots, \Diamond \phi_m \Rightarrow q_1, \ldots, q_k, \Diamond \psi_1, \ldots, \Diamond \psi_k \]
such a sequent is valid if and only if

either \( p_i = q_j \) for some \( i \) and \( j \)

or \( \phi_i \Rightarrow \psi_1, \ldots, \psi_k \) is valid for some \( i \in \{1, \ldots, m\} \)

transformation of sequents: termination

we can apply the transformation rules in any order

does the transformation always terminate?

intuitively yes, because the number of connectives decreases in every step

can the order in which the transformations are performed affect the result?

intuitively no, because the transformations are independent

(such intuitive arguments need to be completed and formalized)
overview

- sequents
- tableaux
- finite models
- standard translation

decidability using semantic tableaux

due to Evert Beth (1908–1964)

method similar to analyze sequents
analyze the nature of possible models via the structure of the formula
a tableau is a finite tree of sequents
if all solid branches close this yields validity of initial sequent
if at least one branch does not close this yields a counterexample

tableaux: propositional part

A, \neg \phi \cdot B \text{ gets successor } A \cdot \phi, B

A \cdot \neg \phi, B \text{ gets successor } A, \phi \cdot B

A, \phi \land \psi \cdot B \text{ gets successor } A, \phi, \psi \cdot B

A \cdot \phi \land \psi, B \text{ gets successors } A \cdot \phi, B \text{ and } A \cdot \psi, B

A, \phi \lor \psi \cdot B \text{ gets successors } A, \phi \cdot B \text{ and } A, \psi \cdot B

A \cdot \phi \lor \psi, B \text{ gets successor } A \cdot \phi, \psi, B

p_1, \ldots, p_n \cdot q_1, \ldots, q_m \text{ either closes or yields a countermodel}
if one branch does not close this yields a countermodel

extension to modal logic

p_1, \ldots, p_n, \diamond \phi_1, \ldots, \diamond \phi_m \cdot q_1, \ldots, q_k, \diamond \psi_1, \ldots, \psi_k

either closes,
or gives a valuation p_1, \ldots, p_n \text{ all true for this world, and gets } m \text{ successors}
\phi_i \cdot \psi_1, \ldots, \psi_k \text{ for every } i \in \{1, \ldots, m\}
stuck if } m = 0
if none of the } m \text{ branches closes this yields a countermodel
\( \Box(p \to q) \to (\Box p \to \Box q) \)

**finite model property: rough idea**

suppose \( \phi \) is satisfiable  
then \( \phi \) is satisfiable in a model with at most \( f(\phi) \) worlds  
so we have a decision procedure for satisfiability of \( \phi \):  
compute \( f(\phi) \)  
consider all (finitely many) models with at most \( f(\phi) \) worlds, 
up to isomorphism  
see whether in one of them \( \phi \) holds  
\( f(\phi) \) is \( 2^{s(\phi)} \) with \( s(\phi) \) the number of subformulas of \( \phi \)
standard translation: idea

translate being true in world \( x \) as predicate \( P \) holds for \( x \)

translate the accessibility relation \( R \) as a binary predicate \( R \)

standard translation: more precisely

from basic modal logic to first-order predicate logic

\[
\begin{align*}
ST_x(p) &= P_x \\
ST_x(\bot) &= \bot \\
ST_x(\neg \phi) &= \neg ST_x(\phi) \\
ST_x(\phi \land \psi) &= ST_x(\phi) \land ST_x(\psi) \\
ST_x(\lozenge \phi) &= \exists y (Rxy \land ST_y(\phi)) \\
ST_x(\square \phi) &= \forall y (Rxy \rightarrow ST_y(\phi))
\end{align*}
\]

standard translation: using only two variables

\[
\begin{align*}
ST_a(\lozenge \phi) &= \exists b (Rab \land ST_b(\phi)) \\
ST_a(\square \phi) &= \forall b (Rab \rightarrow ST_b(\phi)) \\
ST_b(\lozenge \phi) &= \exists a (Rba \land ST_a(\phi)) \\
ST_b(\square \phi) &= \forall a (Rba \rightarrow ST_a(\phi))
\end{align*}
\]
decidability via translation

the translation preserves satisfiability

\( M, w \models \phi \) if and only if \( \text{ST}_{\lambda}(\phi)[x := w] \) is satisfiable

first-order predicate logic using only two variables is decidable