advanced logic
2019 02 21
lecture 6

overview

- sequents
- tableaux
- finite models
- standard translation

overview

- sequents
- tableaux
- finite models
- standard translation

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 $A, \phi \Rightarrow B$

$A, \phi \wedge \psi \Rightarrow B$ if and only if $A, \phi, \psi \Rightarrow B$

$A \Rightarrow \phi \wedge \psi, B$ if and only if both $A \Rightarrow \phi, B$ and $A \Rightarrow \psi, B$

$A, \phi \vee \psi \Rightarrow B$ if and only if both $A, \phi \Rightarrow B$ and $A, \psi \Rightarrow B$

$A \Rightarrow \phi \vee \psi, B$ if and only if $A \Rightarrow \phi, \psi, B$

gives a decision procedure for propositional logic: ϕ valid iff $\Rightarrow \phi$ valid

intuition and notation

we work in sequent calculus

we start with the intended conclusion $\vdash \phi$

we try to build a proof while moving upwards

in fact we use: formula is valid iff it is derivable in sequent calculus

notation guidelines: 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

$\phi = (p \wedge q) \to (p \vee q) \equiv \neg(p \wedge q) \vee (p \vee q)$

$\Rightarrow \neg(p \wedge q) \vee (p \vee q)$

$\Rightarrow \neg(p \wedge q), (p \vee q)$

$p \wedge q \Rightarrow p \vee q$

$p, q \Rightarrow p \vee q$

$p, q \Rightarrow p, q$

the sequent is valid so ϕ is valid

example

$\phi = (p \vee q) \to (p \wedge q) \equiv \neg(p \vee q) \vee (p \wedge q)$

$\Rightarrow \neg(p \vee q) \vee (p \wedge q)$

$\Rightarrow \neg(p \vee q), (p \wedge q)$

$p \vee q \Rightarrow p \wedge q$

$p \Rightarrow p \wedge q$ and $q \Rightarrow p \wedge q$

$p \Rightarrow p$ and $p \Rightarrow q$ and $q \Rightarrow p$ and $q \Rightarrow q$

the sequent is not valid so ϕ is not valid

transformation of sequents: intuition

$p_1, \ldots, p_n, \Diamond\phi_1, \ldots, \Diamond\phi_m \Rightarrow q_1, \ldots, q_k, \Diamond\psi_1, \ldots, \Diamond\psi_k$

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

extension to modal logic

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 $i \in \{1, \ldots, m\}$

decidability using sequents

rewrite formula

rewrite sequent

we may need to rewrite a formula again

decide on validity of sequent

conclude on valitity of formula

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)

example

$\Box(p \to q) \to (\Box p \to \Box q)$

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: proposional part

$A, \neg\phi \bullet B$ gets successor $A \bullet \phi, B$

$A \bullet \neg\phi, B$ gets successor $A, \phi \bullet B$

$A, \phi \wedge \psi \bullet B$ gets successor $A, \phi, \psi \bullet B$

$A \bullet \phi \wedge \psi, B$ gets successors $A \bullet \phi, B$ and $A \bullet \psi, B$

$A, \phi \vee \psi \bullet B$ gets successors $A, \phi \bullet B$ and $A, \psi \bullet B$

$A \bullet \phi \vee \psi, B$ gets successor $A \bullet \phi, \psi, B$

$p_1, \ldots, p_n \bullet q_1, \ldots, q_m$ either closes or yields a countermodel

if one branch does not close this yields a countermodel

extension to modal logic

$p_1, \ldots, p_n, \Diamond\phi_1, \ldots, \Diamond\phi_m \bullet q_1, \ldots, q_k, \Diamond\psi_1, \ldots, \psi_k$

either closes,

or gives a valuation $p_1, \ldots, p_n$ all true for this world,

and gets m (non-solid) successors

$\phi_i \bullet \psi_1, \ldots, \psi_k$ for every $i \in \{1, \ldots, m\}$

stuck if $m = 0$

if none of the m branches closes this yields a countermodel

example

$\Box(p \to q) \to (\Box p \to \Box q)$

overview

- sequents
- tableaux
- finite models
- standard translation

finite model property

definition finite model property:
if ϕ is satisfiable then ϕ is satisfiable on a finite model

(basic) modal logic has the finite model theory

suppose ϕ is satisfiable in $\mathcal{M}_0, x_0$

make the tree unravelling of $\mathcal{M}_0$ at x_0;
we get tree model $\mathcal{M}_1$ with root x_1

$\mathcal{M}_1, x_1$ and $\mathcal{M}_0, x_0$ are bisimilar

hence (using a theorem!) they are modally equivalent

hence ϕ is satisfiable in $\mathcal{M}_1, x_1$

restrict the length of the branches up to the modal depth of ϕ

restrict the amount of branches by analyzing ϕ

effective finite model property

suppose ϕ is satisfiable

then ϕ is satisfiable in a model with at most $f(\phi)$ worlds

so we have a decision procedure for satisfiability of ϕ:

compute $f(\phi)$

consider all (finitely many) models with at most $f(\phi)$ worlds,
up to isomorphism

see whether in one of them ϕ holds

$f(\phi)$ can be $2^{s(\phi)}$ with $s(\phi)$ the number of subformulas of ϕ

overview

- sequents
- tableaux
- finite models
- standard translation

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