Computational Logic

Vincent van Oostrom
Course/slides by Aart Middeldorp

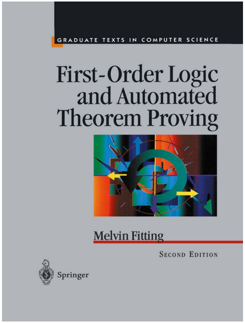
Department of Computer Science
University of Innsbruck

SS 2020

Organisation

- LVA 703824
- Tuesday 9:15 – 12:00 and Thursday 13:15 – 15:00 in 3W03
- http://cl-informatik.uibk.ac.at/teaching/ss20/cl/
- consultation hours: Monday 15:00 – 16:30 in 3M12

Organisation


GRADUATE TEXTS IN COMPUTER SCIENCE
First-Order Logic and Automated Theorem Proving
Melvin Fitting
Second Edition
Springer

Literature

Melvin Fitting
First-Order Logic and Automated Theorem Proving, 2nd edition, Springer-Verlag, 1996

Online Material

slides are available from uibk.ac.at domain

Schedule

week 1	March 3	week 4	March 24	week 7	April 28
week 2	March 10	week 5	March 31	week 8	May 5
week 3	March 17	week 6	April 21	week 9	May 12 (exam)

Outline

- Organisation
- Content
- Propositional Logic
- Semantic Tableaux
- Further Reading

Part I: Propositional Logic

compactness, completeness, Hilbert systems, Hintikka's lemma, interpolation, logical consequence, model existence theorem, propositional semantic tableaux, soundness

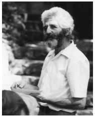
Part II: First-Order Logic

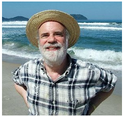
compactness, completeness, Craig's interpolation theorem, cut elimination, first-order semantic tableaux, Herbrand models, Hilbert systems, Hintikka's lemma, Löwenheim–Skolem, logical consequence, model existence theorem, prenex form, skolemization, soundness

Part III: Limitations and Extensions of First-Order Logic

Curry–Howard isomorphism, intuitionistic logic, Kripke models, second-order logic, simply-typed λ-calculus

Content





Outline

- Organisation
- Content
- Propositional Logic
  - Syntax
  - Semantics
  - Replacement Theorem
  - Uniform Notation
  - Normal Forms
- Semantic Tableaux
- Further Reading

Definition

(propositional) atomic formula is propositional letter, $\top$ or $\bot$

Definition

set of propositional formulas is smallest set $\mathbf{P}$ such that

- if A is atomic formula then $A \in \mathbf{P}$
- if $X \in \mathbf{P}$ then $\neg X \in \mathbf{P}$
- if $\circ$ is binary symbol and $X, Y \in \mathbf{P}$ then $(X \circ Y) \in \mathbf{P}$

Theorem (Principle of Structural Induction)

every formula of propositional formula has property Q provided

- *basis step*

  every atomic formula has property Q

- *induction steps*

  if X has property Q then $\neg X$ has property Q

  if X and Y have property Q then $X \circ Y$ has property Q

Theorem (Principle of Structural Recursion)

there exists unique function f defined on **P** *such that*

- *basis step*

  value of f is specified explicitly on atomic formulas

- *induction steps*

  value of f on $\neg X$ is specified in terms of value of f on X

  value of f on $X \circ Y$ is specified in terms of values of f on X and on Y

Definition

immediate subformulas are defined as follows:

- atomic formula has no immediate subformulas
- only immediate subformula of $\neg X$ is X
- immediate subformulas of $(X \circ Y)$ are X and Y

Definitions

- set of subformulas of formula X is smallest set S that contains X and, for every member Y of S, all immediate subformulas of Y
- X is improper subformula of X

Outline

- Organisation
- Content
- Propositional Logic
  - Syntax
  - Semantics
  - Replacement Theorem
  - Uniform Notation
  - Normal Forms
- Semantic Tableaux
- Further Reading

Definitions

- 2 truth values: t and f
- 16 different two-place functions from $\{\,t, f\,\}$ to $\{\,t, f\,\}$
- 8 primary connectives and 2 secondary connectives

		$\wedge$	$\vee$	$\supset$	$\subset$	$\uparrow$	$\downarrow$	$\not\supset$	$\not\subset$
t	t	t	t	t	t	f	f	f	f
t	f	f	t	f	t	t	f	t	f
f	t	f	t	t	f	t	f	f	t
f	f	f	f	t	t	t	t	f	f

		$\equiv$	$\not\equiv$
t	t	t	f
t	f	f	t
f	t	f	t
f	f	t	f

Definition

propositional formula X is tautology if $v(X) = \mathrm{t}$ for every valuation v

Definition

set S of propositional formulas is satisfiable if some valuation maps every member of S to t

Definition

for binary operations $\circ$ and $\bullet$ on $\{\,\mathrm{t}, \mathrm{f}\,\}$: $\circ$ is dual of $\bullet$ if $\neg(x \circ y) = (\neg x \bullet \neg y)$

Examples

$\wedge$ is dual of $\vee$ $\downarrow$ is dual of $\uparrow$ $\not\subset$ is dual of $\supset$

Definition

for propositional formula X we write X^d for result of replacing

- every occurrence of $\top$ with occurrence of $\bot$
- every occurrence of $\bot$ with occurrence of $\top$
- every occurrence of binary symbol with occurrence of its dual

Outline

- Organisation
- Content
- Propositional Logic
  - Syntax
  - Semantics
  - Replacement Theorem
  - Uniform Notation
  - Normal Forms
- Semantic Tableaux
- Further Reading

Theorem

given propositional formulas $F(P)$, X and Y, valuation v

if $v(X) = v(Y)$ then $v(F(X)) = v(F(Y))$

Theorem

if $X \equiv Y$ is tautology then so is $F(X) \equiv F(Y)$

Definition

propositional formula X is in negation normal form if negation symbols $\neg$ occur only in front of propositional letters

Lemma

every propositional formula can be put into negation normal form

Example

$$
\begin{aligned}
\neg[(P \supset Q) \wedge (R \uparrow (\neg P \wedge Q))] &\equiv \neg(P \supset Q) \vee \neg(R \uparrow (\neg P \wedge Q)) \\
&\equiv (\neg P \not\subset \neg Q) \vee \neg(R \uparrow (\neg P \wedge Q)) \\
&\equiv (\neg P \not\subset \neg Q) \vee (\neg R \downarrow \neg(\neg P \wedge Q)) \\
&\equiv (\neg P \not\subset \neg Q) \vee (\neg R \downarrow (\neg\neg P \vee \neg Q)) \\
&\equiv (\neg P \not\subset \neg Q) \vee (\neg R \downarrow (P \vee \neg Q))
\end{aligned}
$$

Outline

- Organisation
- Content
- Propositional Logic
  - Syntax
  - Semantics
  - Replacement Theorem
  - Uniform Notation
  - Normal Forms
- Semantic Tableaux
- Further Reading

Definition

conjunctive			disjunctive		
α	α_1	α_2	β	β_1	β_2
$X \wedge Y$	X	Y	$\neg(X \wedge Y)$	$\neg X$	$\neg Y$
$\neg(X \vee Y)$	$\neg X$	$\neg Y$	$X \vee Y$	X	Y
$\neg(X \supset Y)$	X	$\neg Y$	$X \supset Y$	$\neg X$	Y
$\neg(X \subset Y)$	$\neg X$	Y	$X \subset Y$	X	$\neg Y$
$\neg(X \uparrow Y)$	X	Y	$X \uparrow Y$	$\neg X$	$\neg Y$
$X \downarrow Y$	$\neg X$	$\neg Y$	$\neg(X \downarrow Y)$	X	Y
$X \not\supset Y$	X	$\neg Y$	$\neg(X \not\supset Y)$	$\neg X$	Y
$X \not\subset Y$	$\neg X$	Y	$\neg(X \not\subset Y)$	X	$\neg Y$

Lemma

for every valuation v and all α- and β-formulas

$$v(\alpha) = v(\alpha_1) \wedge v(\alpha_2) \qquad v(\beta) = v(\beta_1) \vee v(\beta_2)$$

Corollary

for every α and β: $\alpha \equiv (\alpha_1 \wedge \alpha_2)$ *and* $\beta \equiv (\beta_1 \vee \beta_2)$ *are tautologies*

Theorem (Principle of Structural Induction)

every formula of propositional logic has property Q *provided*

- *basis step*

  every atomic formula and its negation has property Q

- *induction steps*

  if X *has property* Q *then* $\neg\neg X$ *has property* Q

  if α_1 *and* α_2 *have property* Q *then* α *has property* Q

  if β_1 *and* β_2 *have property* Q *then* β *has property* Q

Definition

rank $r(X)$ of propositional formula is defined as follows:

- $r(A) = r(\neg A) = r(\top) = r(\bot) = 0$
- $r(\neg\top) = r(\neg\bot) = 1$
- $r(\neg\neg Z) = r(Z) + 1$
- $r(\alpha) = r(\alpha_1) + r(\alpha_2) + 1$
- $r(\beta) = r(\beta_1) + r(\beta_2) + 1$

Example

$$\begin{aligned} &r(\neg[(P \supset Q) \wedge (R \uparrow (\neg P \wedge Q))]) \\ &\quad = r(\neg(P \supset Q)) + r(\neg(R \uparrow (\neg P \wedge Q)) + 1 \\ &\quad = r(P) + r(\neg Q) + 1 + r(R) + r(\neg P \wedge Q) + 1 + 1 \\ &\quad = r(P) + r(\neg Q) + 1 + r(R) + r(\neg P) + r(Q) + 1 + 1 + 1 \\ &\quad = 4 \end{aligned}$$

Outline

- Organisation
- Content
- Propositional Logic
  - Syntax
  - Semantics
  - Replacement Theorem
  - Uniform Notation
  - Normal Forms
- Semantic Tableaux
- Further Reading

Definitions

given list $X_1, \ldots, X_n$ of propositional formulas

- $[X_1, \ldots, X_n]$ is generalized disjunction of $X_1, \ldots, X_n$
- $\langle X_1, \ldots, X_n \rangle$ is generalized conjunction of $X_1, \ldots, X_n$
- $v([X_1, \ldots, X_n]) = \begin{cases} \text{t} & \text{if } v(X_i) = \text{t for some } i \in \{1, \ldots, n\} \\ \text{f} & \text{otherwise} \end{cases}$
- $v(\langle X_1, \ldots, X_n \rangle) = \begin{cases} \text{t} & \text{if } v(X_i) = \text{t for all } i \in \{1, \ldots, n\} \\ \text{f} & \text{otherwise} \end{cases}$

Definitions

- literal is propositional letter or negation of propositional letter or $\top$ or $\bot$
- clause is disjunction $[X_1, \ldots, X_n]$ consisting of literals $X_1, \ldots, X_n$
- dual clause is conjunction $\langle X_1, \ldots, X_n \rangle$ consisting of literals $X_1, \ldots, X_n$

Definitions

- propositional formula is in conjunctive normal form or in clause form if it is conjunction $\langle C_1, \dots, C_n \rangle$ of clauses
- propositional formula is in disjunctive normal form or in dual clause form if it is disjunction $[D_1, \dots, D_n]$ of dual clauses

Theorem (Normal Form)

there are algorithms for converting propositional formula into clause form and into dual clause form

Proof (clause form)

- step 1

  start with $\langle [X] \rangle$

  $\dots$

  if $\langle D_1, \dots, D_k \rangle$ is not yet conjunctive normal form continue with

- step $n + 1$

  select D_i which contains non-literal N

  - if $N = \neg\top$ replace N with $\bot$
  - if $N = \neg\bot$ replace N with $\top$
  - if $N = \neg\neg Z$ replace N with Z
  - if N is β-formula replace N with β_1 and β_2
  - if N is α-formula replace disjunction D_i with two disjunctions:
    - one with α replaced by α_1
    - one with α replaced by α_2

Clause Set Reduction Rules

$$\frac{\neg\top}{\bot} \qquad \frac{\neg\bot}{\top} \qquad \frac{\neg\neg Z}{Z} \qquad \frac{\beta}{\begin{array}{c}\beta_1\\ \beta_2\end{array}} \qquad \frac{\alpha}{\alpha_1 \mid \alpha_2}$$

Lemma

if S is conjunction of disjunctions and S' is obtained from S by applying one clause set reduction rule then $S \equiv S'$ is tautology

Clause Form Algorithm

```
let $S$ be $\langle [X] \rangle$
while some member of $S$ contains non-literal do
    select member $D$ of $S$ containing non-literal
    select non-literal $N$ of $D$
    apply appropriate clause set reduction rule to $N$ in $D$, producing new $S$
```

Theorem

Clause Form Algorithm terminates and produces clause form S such that $S \equiv X$ is tautology

Example

$$
\begin{aligned}
&\langle [\,(P \supset (Q \supset R)) \supset ((P \supset Q) \supset (P \supset R))\,] \rangle \\
&\qquad \equiv \langle [\,\neg(P \supset (Q \supset R)), (P \supset Q) \supset (P \supset R)\,] \rangle \\
&\qquad \equiv \langle [\,\neg(P \supset (Q \supset R)), \neg(P \supset Q), P \supset R\,] \rangle \\
&\qquad \equiv \langle [\,\neg(P \supset (Q \supset R)), \neg(P \supset Q), \neg P, R\,] \rangle \\
&\qquad \equiv \langle [\,P, \neg(P \supset Q), \neg P, R\,], [\,\neg(Q \supset R), \neg(P \supset Q), \neg P, R\,] \rangle \\
&\qquad \equiv \langle [\,P, P, \neg P, R\,], [\,P, \neg Q, \neg P, R\,], [\,\neg(Q \supset R), \neg(P \supset Q), \neg P, R\,] \rangle \\
&\qquad \equiv \langle [\,P, P, \neg P, R\,], [\,P, \neg Q, \neg P, R\,], [\,Q, \neg(P \supset Q), \neg P, R\,], \\
&\qquad\qquad [\,\neg R, \neg(P \supset Q), \neg P, R\,] \rangle \\
&\qquad \equiv \langle [\,P, P, \neg P, R\,], [\,P, \neg Q, \neg P, R\,], [\,Q, P, \neg P, R\,], [\,Q, \neg Q, \neg P, R\,], \\
&\qquad\qquad [\,\neg R, \neg(P \supset Q), \neg P, R\,] \rangle \\
&\qquad \equiv \langle [\,P, P, \neg P, R\,], [\,P, \neg Q, \neg P, R\,], [\,Q, P, \neg P, R\,], [\,Q, \neg Q, \neg P, R\,], \\
&\qquad\qquad [\,\neg R,\ P,\ \neg P,\ R\,], [\,\neg R,\ \neg Q,\ \neg P,\ R\,] \rangle
\end{aligned}
$$

Theorem

Clause Form Algorithm terminates and produces clause form S such that $S \equiv X$ is tautology

Proof Sketch

rank of generalized disjunction $[X_1, \dots, X_n]$ is $r(X_1) + \cdots + r(X_n)$

incrementally build tree whose leaves correspond to ranks of generalized disjunctions in current S:

- root node with label r($[X]$)
- employed clause set reduction rule determines tree expansion
- conclude by König's Lemma

Dual Clause Set Reduction Rules

$$\frac{\neg\bot}{\top} \qquad \frac{\neg\top}{\bot} \qquad \frac{\neg\neg Z}{Z} \qquad \frac{\alpha}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}} \qquad \frac{\beta}{\beta_1 \mid \beta_2}$$

Example

$$\begin{aligned}
[\langle (P \downarrow Q) \supset (Q \vee \neg(P \vee \neg Q))\rangle] &\equiv [\langle \neg(P \downarrow Q)\rangle, \langle Q \vee \neg(P \vee \neg Q)\rangle] \\
&\equiv [\langle P\rangle, \langle Q\rangle, \langle Q \vee \neg(P \vee \neg Q)\rangle] \\
&\equiv [\langle P\rangle, \langle Q\rangle, \langle Q\rangle, \langle \neg(P \vee \neg Q)\rangle] \\
&\equiv [\langle P\rangle, \langle Q\rangle, \langle Q\rangle, \langle \neg P, \neg\neg Q\rangle] \\
&\equiv [\langle P\rangle, \langle Q\rangle, \langle Q\rangle, \langle \neg P, Q\rangle]
\end{aligned}$$

Dual Clause Form Algorithm

```
let $S$ be $[\langle X \rangle]$
while some member of $S$ contains non-literal do
    select member $C$ of $S$ containing non-literal
    select non-literal $N$ of $C$
    apply appropriate dual clause set reduction rule to $N$ in $C$, producing new $S$
```

Lemma

if S is disjunction of conjunctions and S' is obtained from S by applying one dual clause set reduction rule then $S \equiv S'$ is tautology

Theorem

Dual Clause Form Algorithm terminates and produces dual clause form S such that $S \equiv X$ is tautology

Outline

- Organisation
- Content
- Propositional Logic
- Semantic Tableaux
  - Definitions
- Further Reading

Tableau Expansion Rules

$$\frac{\neg\neg Z}{Z} \qquad \frac{\neg\bot}{\top} \qquad \frac{\neg\top}{\bot} \qquad \frac{\alpha}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}} \qquad \frac{\beta}{\beta_1 \mid \beta_2}$$

Definition

finite set $\{A_1, \ldots, A_n\}$ of propositional formulas

1. following one-branch tree is tableau for $\{A_1, \ldots, A_n\}$:

$$\begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_n \end{array}$$

2. if T is tableau for $\{A_1, \ldots, A_n\}$ and T^* results from T by application of tableau expansion rule then T^* is tableau for $\{A_1, \ldots, A_n\}$

Outline

- Organisation
- Content
- Propositional Logic
- Semantic Tableaux
- Further Reading

Fitting

- Chapter 1
- Chapter 2 (except for Section 2.9)
- Section 3.1 !

Computational Logic

Vincent van Oostrom
Course/slides by Aart Middeldorp

Department of Computer Science
University of Innsbruck

SS 2020

Outline

- Summary of Previous Lecture
- Semantic Tableaux
- Hintikka's Lemma
- Model Existence Theorem
- Exercises
- Further Reading

Definition

8 primary connectives and 2 secondary connectives

		$\wedge$	$\vee$	$\supset$	$\subset$	$\uparrow$	$\downarrow$	$\not\supset$	$\not\subset$
t	t	t	t	t	t	f	f	f	f
t	f	f	t	f	t	t	f	t	f
f	t	f	t	t	f	t	f	f	t
f	f	f	f	t	t	t	t	f	f

		$\equiv$	$\not\equiv$
t	t	t	f
t	f	f	t
f	t	f	t
f	f	t	f

Definition

propositional formula X is tautology if $v(X) = \mathsf{t}$ for every valuation v

Definition

set S of propositional formulas is satisfiable if some valuation maps every member of S to t

Definition

for binary operations $\circ$ and $\bullet$ on $\{t, f\}$: $\circ$ is dual of $\bullet$ if $\neg(x \circ y) = (\neg x \bullet \neg y)$

Definition (Uniform Notation)

conjunctive			disjunctive		
α	α_1	α_2	β	β_1	β_2
$X \wedge Y$	X	Y	$\neg(X \wedge Y)$	$\neg X$	$\neg Y$
$\neg(X \vee Y)$	$\neg X$	$\neg Y$	$X \vee Y$	X	Y
$\neg(X \supset Y)$	X	$\neg Y$	$X \supset Y$	$\neg X$	Y
$\neg(X \subset Y)$	$\neg X$	Y	$X \subset Y$	X	$\neg Y$
$\neg(X \uparrow Y)$	X	Y	$X \uparrow Y$	$\neg X$	$\neg Y$
$X \downarrow Y$	$\neg X$	$\neg Y$	$\neg(X \downarrow Y)$	X	Y
$X \not\supset Y$	X	$\neg Y$	$\neg(X \not\supset Y)$	$\neg X$	Y
$X \not\subset Y$	$\neg X$	Y	$\neg(X \not\subset Y)$	X	$\neg Y$

Lemma

for every valuation v and all α- and β-formulas

$$v(\alpha) = v(\alpha_1) \land v(\alpha_2) \qquad v(\beta) = v(\beta_1) \lor v(\beta_2)$$

Corollary

for every α and β: $\alpha \equiv (\alpha_1 \land \alpha_2)$ and $\beta \equiv (\beta_1 \lor \beta_2)$ are tautologies

Definition

rank $r(X)$ of propositional formula is defined as follows:

- $r(A) = r(\neg A) = r(\top) = r(\bot) = 0$
- $r(\neg\top) = r(\neg\bot) = 1$
- $r(\neg\neg Z) = r(Z) + 1$
- $r(\alpha) = r(\alpha_1) + r(\alpha_2) + 1$
- $r(\beta) = r(\beta_1) + r(\beta_2) + 1$

Definitions

given list $X_1, \ldots, X_n$ of propositional formulas

- $[X_1, \ldots, X_n]$ is generalized disjunction of $X_1, \ldots, X_n$
- $\langle X_1, \ldots, X_n \rangle$ is generalized conjunction of $X_1, \ldots, X_n$

Definitions

- literal is propositional letter or negation of propositional letter or $\top$ or $\bot$
- clause is disjunction $[X_1, \ldots, X_n]$ consisting of literals $X_1, \ldots, X_n$
- dual clause is conjunction $\langle X_1, \ldots, X_n \rangle$ consisting of literals $X_1, \ldots, X_n$
- propositional formula is in conjunctive normal form or in clause form if it is conjunction $\langle C_1, \ldots, C_n \rangle$ of clauses
- propositional formula is in disjunctive normal form or in dual clause form if it is disjunction $[D_1, \ldots, D_n]$ of dual clauses

Clause Set Reduction Rules

$$\frac{\neg\top}{\bot} \qquad \frac{\neg\bot}{\top} \qquad \frac{\neg\neg Z}{Z} \qquad \frac{\beta}{\begin{array}{c}\beta_1\\ \beta_2\end{array}} \qquad \frac{\alpha}{\alpha_1 \mid \alpha_2}$$

Clause Form Algorithm

```
let $S$ be $\langle[X]\rangle$
while some member of $S$ contains non-literal do
    select member $D$ of $S$ containing non-literal
    select non-literal $N$ of $D$
    apply appropriate clause set reduction rule to $N$ in $D$, producing new $S$
```

Theorem

Clause Form Algorithm terminates and produces clause form S such that $S \equiv X$ is tautology

Dual Clause Set Reduction Rules

$$\frac{\neg\bot}{\top} \qquad \frac{\neg\top}{\bot} \qquad \frac{\neg\neg Z}{Z} \qquad \frac{\alpha}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}} \qquad \frac{\beta}{\beta_1 \mid \beta_2}$$

Dual Clause Form Algorithm

```
let $S$ be $[\langle X \rangle]$
while some member of $S$ contains non-literal do
    select member $C$ of $S$ containing non-literal
    select non-literal $N$ of $C$
    apply appropriate dual clause set reduction rule to $N$ in $C$, producing new $S$
```

Theorem

Dual Clause Form Algorithm terminates and produces dual clause form S such that $S \equiv X$ is tautology

Tableau Expansion Rules

$$\frac{\neg\neg Z}{Z} \qquad \frac{\neg\bot}{\top} \qquad \frac{\neg\top}{\bot} \qquad \frac{\alpha}{\begin{matrix}\alpha_1\\ \alpha_2\end{matrix}} \qquad \frac{\beta}{\beta_1 \mid \beta_2}$$

Definition

finite set $\{A_1, \ldots, A_n\}$ of propositional formulas

1. following one-branch tree is tableau for $\{A_1, \ldots, A_n\}$:

$$\begin{matrix} A_1 \\ A_2 \\ \vdots \\ A_n \end{matrix}$$

2. if T is tableau for $\{A_1, \ldots, A_n\}$ and T^* results from T by application of tableau expansion rule then T^* is tableau for $\{A_1, \ldots, A_n\}$

Part I: Propositional Logic

compactness, completeness, Hilbert systems, Hintikka's lemma, interpolation, logical consequence, model existence theorem, propositional semantic tableaux, soundness

Part II: First-Order Logic

compactness, completeness, Craig's interpolation theorem, cut elimination, first-order semantic tableaux, Herbrand models, Herbrand's theorem, Hilbert systems, Hintikka's lemma, Löwenheim-Skolem, logical consequence, model existence theorem, prenex form, skolemization, soundness

Part III: Limitations and Extensions of First-Order Logic

Curry-Howard isomorphism, intuitionistic logic, Kripke models, second-order logic, simply-typed λ-calculus

Outline

- Summary of Previous Lecture
- Semantic Tableaux
  - Definitions
  - Soundness
- Hintikka's Lemma
- Model Existence Theorem
- Exercises
- Further Reading

Tableau Expansion Rules

$$\frac{\neg\neg Z}{Z} \qquad \frac{\neg\bot}{\top} \qquad \frac{\neg\top}{\bot} \qquad \frac{\alpha}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}} \qquad \frac{\beta}{\beta_1 \mid \beta_2}$$

Definition

finite set $\{A_1, \dots, A_n\}$ of propositional formulas

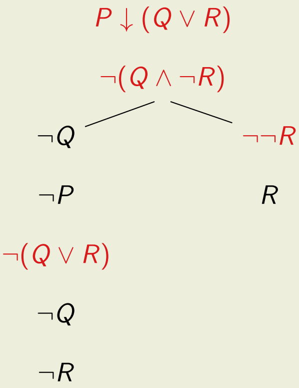
1. following one-branch tree is tableau for $\{A_1, \dots, A_n\}$:

$$\begin{array}{c} A_1 \\ A_2 \\ \vdots \\ A_n \end{array}$$

2. if T is tableau for $\{A_1, \dots, A_n\}$ and T^* results from T by application of tableau expansion rule then T^* is tableau for $\{A_1, \dots, A_n\}$

Example

tableau for $\{P \downarrow (Q \vee R), \neg(Q \wedge \neg R)\}$:



$P \downarrow (Q \vee R)$

$\neg(Q \wedge \neg R)$

$\neg Q$ $\neg\neg R$

$\neg P$ R

$\neg(Q \vee R)$

$\neg Q$

$\neg R$

Definitions

- branch θ of tableau is closed if both X and $\neg X$ occur on θ for some propositional formula X, or if $\bot$ occurs on θ
- tableau is closed if every branch is closed

Definitions

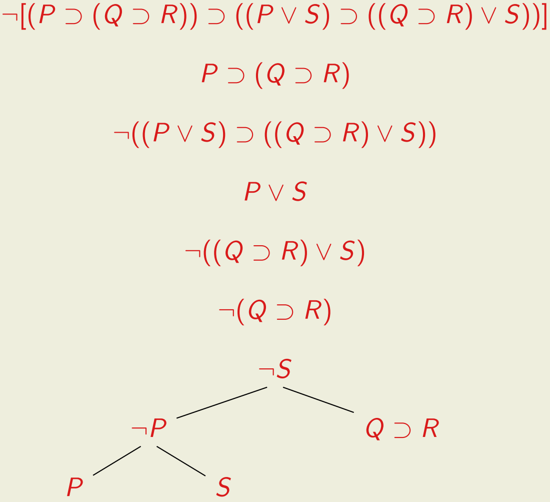
- tableau proof of X is closed tableau for $\{\neg X\}$
- X is theorem if X has tableau proof, denoted by $\vdash_{pt} X$

Definitions

- branch θ of tableau is atomically closed if both A and $\neg A$ occur on θ for some propositional letter A, or if $\bot$ occurs on θ
- tableau is atomically closed if every branch is atomically closed

Example

tableau proof of $(P \supset (Q \supset R)) \supset ((P \vee S) \supset ((Q \supset R) \vee S))$:



$\neg[(P \supset (Q \supset R)) \supset ((P \vee S) \supset ((Q \supset R) \vee S))]$

$P \supset (Q \supset R)$

$\neg((P \vee S) \supset ((Q \supset R) \vee S))$

$P \vee S$

$\neg((Q \supset R) \vee S)$

$\neg(Q \supset R)$

$\neg S$

$\neg P$

$Q \supset R$

P

S

(non-atomically) closed

Definition

tableau is strict if no formula has had Tableau Expansion Rule applied to it twice on same branch

Example

two tableau proofs of $X = (P \land (Q \supset (R \lor S))) \supset (P \lor Q)$:

$$\neg X$$

$$P \land (Q \supset (R \lor S))$$

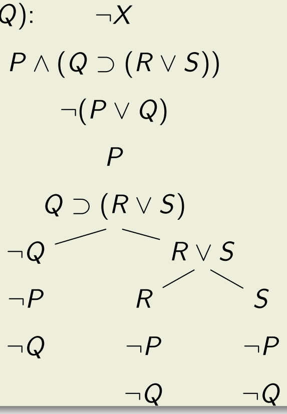
$$\neg(P \lor Q)$$

$$P$$

$$Q \supset (R \lor S)$$

$$\neg P$$

$$\neg Q$$



$\neg X$

$P \land (Q \supset (R \lor S))$

$\neg(P \lor Q)$

P

$Q \supset (R \lor S)$

$\neg Q$ $R \lor S$

$\neg P$ R S

$\neg Q$ $\neg P$ $\neg P$

$\neg Q$ $\neg Q$

Outline

- Summary of Previous Lecture
- Semantic Tableaux
  - Definitions
  - Soundness
- Hintikka's Lemma
- Model Existence Theorem
- Exercises
- Further Reading

Definitions

- set S of propositional formulas is satisfiable if some valuation maps every member of S to t
- tableau branch θ is satisfiable if set of propositional formulas on it is satisfiable
- tableau T is satisfiable if at least one branch of T is satisfiable

Lemma

any application of Tableau Expansion Rule to satisfiable tableau yields another satisfiable tableau

Proof

suppose T is satisfiable tableau and let T^* be obtained by applying Tableau Expansion Rule to formula occurrence X on branch θ

let τ be satisfiable branch of T

- if $\tau \neq \theta$ then τ is (satisfiable) branch of T^*
- if $\tau = \theta$ then case distinction on Tableau Expansion Rule applied to X

1. $X = \neg\neg Z$ or $X = \neg\bot$ or $X = \neg\top$: easy
2. $X = \alpha$: θ is extended with α_1 and α_2 to produce T^*

   $v(\alpha) = \mathrm{t} \quad \Longrightarrow \quad v(\alpha_1) = v(\alpha_2) = \mathrm{t}$

   hence extended branch in T^* is satisfiable
3. $X = \beta$: left and right children were added to last node of θ, one labeled β_1 and one labeled β_2, to produce T^*

   $v(\beta) = \mathrm{t} \quad \Longrightarrow \quad v(\beta_1) = \mathrm{t}$ or $v(\beta_2) = \mathrm{t}$

   hence one of new branches in T^* is satisfiable

Lemma

if S admits closed tableau then S is not satisfiable

Proof (by contradiction)

if S is satisfiable then initial tableau is satisfiable

every subsequent tableau is satisfiable (by previous lemma)

final closed tableau is satisfiable ↯

Theorem (Propositional Tableau Soundness)

if X has tableau proof then X is tautology

Proof

closed tableau for $\{\neg X\}$

$\{\neg X\}$ is not satisfiable (by previous lemma)

X is tautology

Outline

- Summary of Previous Lecture
- Semantic Tableaux
- Hintikka's Lemma
- Model Existence Theorem
- Exercises
- Further Reading

Definition

set **H** of propositional formulas is propositional Hintikka set provided

1. for any propositional letter A, not both $A \in \mathbf{H}$ and $\neg A \in \mathbf{H}$
2. $\bot \notin \mathbf{H}$, $\neg\top \notin \mathbf{H}$
3. if $\neg\neg Z \in \mathbf{H}$ then $Z \in \mathbf{H}$
4. if $\alpha \in \mathbf{H}$ then $\alpha_1 \in \mathbf{H}$ and $\alpha_2 \in \mathbf{H}$
5. if $\beta \in \mathbf{H}$ then $\beta_1 \in \mathbf{H}$ or $\beta_2 \in \mathbf{H}$

Examples

- $\varnothing$ is Hintikka set
- set of all propositional variables is Hintikka set
- $\{P \wedge (\neg Q \supset R), P, (\neg Q \supset R), \neg\neg Q, Q\}$ is Hintikka set

Lemma (Hintikka's Lemma)

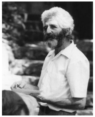
every propositional Hintikka set is satisfiable

Proof

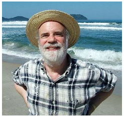
define valuation f for propositional Hintikka set $\mathbf{H}$ as follows:

$$f(A) = \begin{cases} \mathrm{t} & \text{if } A \in \mathbf{H} \\ \mathrm{f} & \text{if } \neg A \in \mathbf{H} \\ \mathrm{f} & \text{otherwise} \end{cases}$$

easy induction proof shows that valuation f maps every member of $\mathbf{H}$ to t



Jaakko Hintikka
(1929 – 2015)



Outline

- Summary of Previous Lecture
- Semantic Tableaux
- Hintikka's Lemma
- Model Existence Theorem
- Exercises
- Further Reading

Definition

collection $\mathcal{C}$ of sets of propositional formulas is propositional consistency property if, for each $S \in \mathcal{C}$:

1. for any propositional letter A, not both $A \in S$ and $\neg A \in S$
2. $\bot \notin S$, $\neg\top \notin S$
3. if $\neg\neg Z \in S$ then $S \cup \{Z\} \in \mathcal{C}$
4. if $\alpha \in S$ then $S \cup \{\alpha_1, \alpha_2\} \in \mathcal{C}$
5. if $\beta \in S$ then $S \cup \{\beta_1\} \in \mathcal{C}$ or $S \cup \{\beta_2\} \in \mathcal{C}$

if $\mathcal{C}$ is propositional consistency property then $S \in \mathcal{C}$ is called $\mathcal{C}$-consistent

Theorem (Propositional Model Existence)

if $\mathcal{C}$ is propositional consistency property and $S \in \mathcal{C}$ then S is satisfiable

Proof (easy case: S is finite)

- enlarge S to member of $\mathcal{C}$ that is Hintikka set:
  - if $\neg\neg Z \in S$ then add Z to S
  - if $\alpha \in S$ then add both α_1 and α_2 to S
  - if $\beta \in S$ then add
    - β_1 to S if $S \cup \{\beta_1\} \in \mathcal{C}$
    - β_2 to S if $S \cup \{\beta_2\} \in \mathcal{C}$
- saturation process terminates because S is finite
- resulting set is Hintikka set and thus satisfiable
- hence subset S is also satisfiable

Definition

propositional consistency property $\mathcal{C}$ is subset closed if, for every $S \in \mathcal{C}$, all subsets of S belong to $\mathcal{C}$

Definition

propositional consistency property $\mathcal{C}$ is of finite character provided $S \in \mathcal{C}$ if and only if every finite subset of S belongs to $\mathcal{C}$

Lemmata

- *every propositional consistency property can be extended to subset closed one*
- *every propositional consistency property of finite character is subset closed*
- *every subset closed propositional consistency property can be extended to one of finite character*

Lemma

every propositional consistency property $\mathcal{C}$ can be extended to subset closed one

Proof

- $\mathcal{C}^+ = \{T \mid T \subseteq S \in \mathcal{C}\}$ is subset closed
- let $T \in \mathcal{C}^+$ so $T \subseteq S$ for some $S \in \mathcal{C}$

1. if $A \in T$ and $\neg A \in T$ then $A \in S$ and $\neg A \in S$
2. if $\bot \in T$ or $\neg\top \in T$ then $\bot \in S$ or $\neg\top \in S$
3. if $\neg\neg Z \in T$ then $\neg\neg Z \in S$ and thus $S \cup \{Z\} \in \mathcal{C}$
   hence $T \cup \{Z\} \in \mathcal{C}^+$
4. if $\alpha \in T$ then $\alpha \in S$ and thus $S \cup \{\alpha_1, \alpha_2\} \in \mathcal{C}$
   hence $T \cup \{\alpha_1, \alpha_2\} \in \mathcal{C}^+$
5. if $\beta \in T$ then $\beta \in S$ and thus $S \cup \{\beta_1\} \in \mathcal{C}$ or $S \cup \{\beta_2\} \in \mathcal{C}$
   hence $T \cup \{\beta_1\} \in \mathcal{C}^+$ or $T \cup \{\beta_2\} \in \mathcal{C}^+$

Lemma

every propositional consistency property $\mathcal{C}$ of finite character is subset closed

Proof

- let $T \subseteq S \in \mathcal{C}$
- all finite subsets of S belong to $\mathcal{C}$
- all finite subsets of T belong to $\mathcal{C}$
- $T \in \mathcal{C}$ because $\mathcal{C}$ is of finite character

Lemma

every subset closed propositional consistency property can be extended to one of finite character

Proof

. . . exercise . . .

Lemma

if $\mathcal{C}$ is propositional consistency property of finite character and $S_1, S_2, S_3, \cdots \in \mathcal{C}$ such that $S_1 \subseteq S_2 \subseteq S_3 \subseteq \cdots$ then $\bigcup_i S_i \in \mathcal{C}$

Proof

it suffices to show that every finite subset $\{A_1, \dots, A_k\}$ of $\bigcup_i S_i$ belongs to $\mathcal{C}$:

- $\forall\ 1 \leqslant i \leqslant k\ \exists\ n_i$ such that $A_i \in S_{n_i}$
- let $N = \max\{n_1, \dots, n_k\}$
- $\{A_1, \dots, A_k\} \subseteq S_N$ and $S_N \in \mathcal{C}$
- $\{A_1, \dots, A_k\} \in \mathcal{C}$ because $\mathcal{C}$ is of finite character

Proof (of Propositional Model Existence Theorem)

given propositional consistency property $\mathcal{C}$ and $S \in \mathcal{C}$

- we may assume that $\mathcal{C}$ is of finite character
- let $X_1, X_2, X_3, \dots$ be enumeration of all propositional formulas
- define sequence $S_1, S_2, S_3, \dots$ of members of $\mathcal{C}$:

$$S_1 = S \qquad\qquad S_{n+1} = \begin{cases} S_n \cup \{X_n\} & \text{if } S_n \cup \{X_n\} \in \mathcal{C} \\ S_n & \text{otherwise} \end{cases}$$

- $S_1 \subseteq S_2 \subseteq S_3 \subseteq \cdots$ and hence $\mathbf{H} = \bigcup_i S_i$ belongs to $\mathcal{C}$ by previous lemma
- $\mathbf{H}$ is maximal in $\mathcal{C}$:
  - suppose $\mathbf{H} \subsetneq K \in \mathcal{C}$ let $X_n \in K \setminus \mathbf{H}$
  - $X_n \notin \mathbf{H}$ and hence $S_n \cup \{X_n\} \notin \mathcal{C}$
  - $S_n \cup \{X_n\} \subseteq K$ ↯
- $\mathbf{H}$ is Hintikka set and hence $S \subseteq \mathbf{H}$ is satisfiable

Outline

- Summary of Previous Lecture
- Semantic Tableaux
- Hintikka's Lemma
- Model Existence Theorem
- Exercises
- Further Reading

Fitting

- Exercise 2.3.2
- Exercise 2.4.3
- Exercise 2.4.12
- Exercise 2.6.1 (imp)
- Bonus: give a translation t of formulas into ones only using conjunction, disjunction and negation, and adapt d to a notion d', such that $r(X) = d'(t(X)$ for all formulas X.
- Exercise 2.6.2
- Bonus: Exercise 2.8.4 (imp)
- Bonus: Exercise 2.8.6 (imp)
- Exercise 2.8.7
- Exercise 3.1.1 !
- Exercise 3.6.3 !

Outline

- Summary of Previous Lecture
- Semantic Tableaux
- Hintikka's Lemma
- Model Existence Theorem
- Exercises
- Further Reading

Fitting

- Section 3.4
- Section 3.5
- Section 3.6 !