universität innsbruck

SS 2024 lecture 1

Logic

Diana Gründlinger **Aart Middeldorp** Fabian Mitterwallner

Alexander Montag Johannes Niederhauser Daniel Rainer

Outline

VO is streamed and recorded

Articify with session ID **0992 9580** for anonymous questions

`ars.uibk.ac.at`

Important Information

- LVA 703026 (VO 3) + 703027 (PS 2)
- `http://cl-informatik.uibk.ac.at/teaching/ss24/lics`
- online registration for VO required until June 30
- OLAT links for VO and PS

Time and Place

VO	Monday	8:30 – 11:00	HSB 1	Aart		
TU	Wednesday	16:15 – 17:00	SR 13	Alexander		
PS	Thursday	12:00 – 13:30	HSB 8	Daniel	group 1	
	Thursday	8:30 – 10:00	HSB 4	Fabian	group 2	(in English)
	Thursday	8:30 – 10:00	SR 12	Johannes	group 3	(in English)
	Thursday	12:00 – 13:30	HS E	Diana	group 4	
	Thursday	13:45 – 15:15	HS 11	Diana	group 5	

PS group change requests until noon tomorrow using SWAp tool

Consultation Hours

Diana Gründlinger	3M03	Thursday 9:30 – 11:00
Aart Middeldorp	3M07	Wednesday 11:30 – 13:00
Fabian Mitterwallner	3M03	Thursday 10:30 – 12:00
Alexander Montag	ÖH Technik	Wednesday 14:00 – 15:00
Johannes Niederhauser	3M03	Friday 9:30 – 11:00
Daniel Rainer	3M03	Wednesday 11:30 – 13:00

Schedule

lecture 1	04.03 & 07.03	lecture 8	06.05 & 16.05
lecture 2	11.03 & 14.03	lecture 9	13.05 & 23.05
lecture 3	18.03 & 21.03	lecture 10	27.05 & 06.06
lecture 4	08.04 & 11.04	lecture 11	03.06 & 06.06
lecture 5	15.04 & 18.04	lecture 12	10.06 & 13.06
lecture 6	22.04 & 25.04	lecture 13	17.06 & 20.06
lecture 7	29.04 & 02.05	lecture 14	24.06 (first exam)

Announcements

- VO is streamed and recorded
- PS in presence, no PS on June 27

Grading — VO

- first exam on June 24
- registration starts 5 weeks before exam and ends 2 weeks before exam
- late registration requests will be ignored
- de-registration is possible until 23:59 on June 20
- second exam on September 20
- third exam on February 26, 2025

Grading — PS

$$\text{score} = \min\left(\tfrac{50}{67}(E+P)+B, 100\right)$$

- E: points for solved exercises (at most 120)
- B: points for bonus exercises (at most 20)
- P: points for presentations of solutions (at most 14)

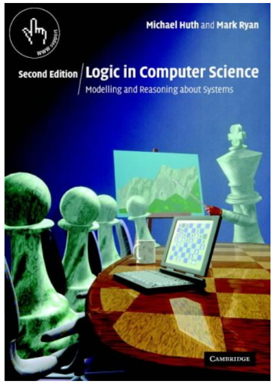
grade: $[0,50) \to \mathbf{5}$ $[50,63) \to \mathbf{4}$ $[63,75) \to \mathbf{3}$ $[75,88) \to \mathbf{2}$ $[88,100] \to \mathbf{1}$

- homework exercises are given on course web site
- solved exercises must be marked in OLAT
- solutions must be uploaded (PDF) in OLAT; deadline: 7 am on Thursday
- 10 points per PS
- two presentations of solutions are mandatory
- 14 points for two presentations; additional presentations give bonus points
- attendance is compulsory; unexcused absence is allowed twice (resulting in 0 points)

Literature

Michael Huth and Mark Ryan
Logic in Computer Science (second edition)
Cambridge University Press, 2004

in Semesterapparat digital version



Online Material

- slides are available on Thursday before lecture on Monday
- solutions to selected exercises are available after they have been discussed in PS

evaluation SS 2023

Evaluation SS 2023 (selected comments)

- Es macht keinen sinn dass der Professor selbst viele Formeln nicht auswendig kann wir aber alle auswendig wissen müssen für den test. Das hat absolut nichts mit Verständnis zu tun. Es is unnötiges auswendig lernen was sich nichts bringt. In jeder anderen situation würde man z.b. die regeln der natural deduction oder Prädikaten logik nachlesen.
- The lecture was interesting, but the slides are very poorly structured or very often something is assumed but not explained why. It's a lot to memorize without the professor explaining/reciting it in a way that the student understands.

  One of the worst lecturers, much is defined as it is, without explanation. And if students attend more different PS, then his subject is the most important thing to him, he doesn't take students into consideration.
- The examples given on the slides are, especially later during the semester, way to simple compared to whats wanted in the proseminar. More examples would be helpful, also during definitions as these can be rather dry otherwise.
- explained an extremely difficult topic very well. the logic course in of itself is complicated and extensive so i can't fault him for that. he knows his stuff for sure.
- This was the best VO in this semester. The slides were good, PS sheets helped a lot.

Outline

1. Introduction

Organisation Motivation Contents

2. Propositional Logic

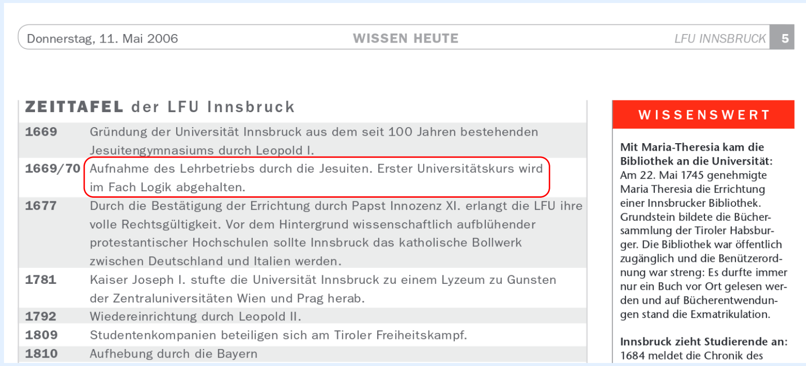
3. Satisfiability and Validity

4. Intermezzo

5. Conjunctive Normal Forms

6. Further Reading

Logic in Innsbruck


Donnerstag, 11. Mai 2006 WISSEN HEUTE LFU INNSBRUCK 5

ZEITTAFEL der LFU Innsbruck

1669	Gründung der Universität Innsbruck aus dem seit 100 Jahren bestehenden Jesuitengymnasiums durch Leopold I.
1669/70	Aufnahme des Lehrbetriebs durch die Jesuiten. Erster Universitätskurs wird im Fach Logik abgehalten.
1677	Durch die Bestätigung der Errichtung durch Papst Innozenz XI. erlangt die LFU ihre volle Rechtsgültigkeit. Vor dem Hintergrund wissenschaftlich aufblühender protestantischer Hochschulen sollte Innsbruck das katholische Bollwerk zwischen Deutschland und Italien werden.
1781	Kaiser Joseph I. stufte die Universität Innsbruck zu einem Lyzeum zu Gunsten der Zentraluniversitäten Wien und Prag herab.
1792	Wiedereinrichtung durch Leopold II.
1809	Studentenkompanien beteiligen sich am Tiroler Freiheitskampf.
1810	Aufhebung durch die Bayern

WISSENSWERT

Mit Maria-Theresia kam die Bibliothek an die Universität: Am 22. Mai 1745 genehmigte Maria Theresia die Errichtung einer Innsbrucker Bibliothek. Grundstein bildete die Büchersammlung der Tiroler Habsburger. Die Bibliothek war öffentlich zugänglich und die Benützerordnung war streng: Es durfte immer nur ein Buch vor Ort gelesen werden und auf Bücherentwendungen stand die Exmatrikulation.

Innsbruck zieht Studierende an: 1684 meldet die Chronik des

Formal Logic at Department of Christian Philosophy

Logic in Computer Science

- During the past 30 years, there has been an extensive and growing interaction between logic and computer science.
- Concepts and methods of logic occupy a central place in computer science, insomuch that logic has been called the calculus of computer science.
- Logic has been much more effective in computer science than it has been in mathematics.

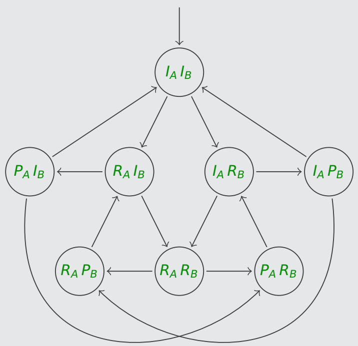
Phokion G. Kolaitis, Moshe Y. Vardi (2001)

Example（数独 Sudoku）

	6		1		4		5	
		8	3		5	6		
2								1
8			4		7			6
		6				3		
7			9		1			4
5								2
		7	2		6	9		
	4		5		8		7	

propositional logic is very useful to quickly develop efficient solver for Sudoku and all kinds of other tasks

Example（Printer Manager）



two users A and B

I_i user i is idle

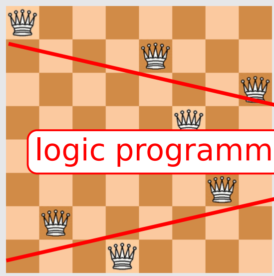
R_i print request by user i

P_i printing document for user i

some questions

- is every P_i preceded by R_i ?
- is every R_i eventually followed by P_i ?

Example（Eight Queens Puzzle）



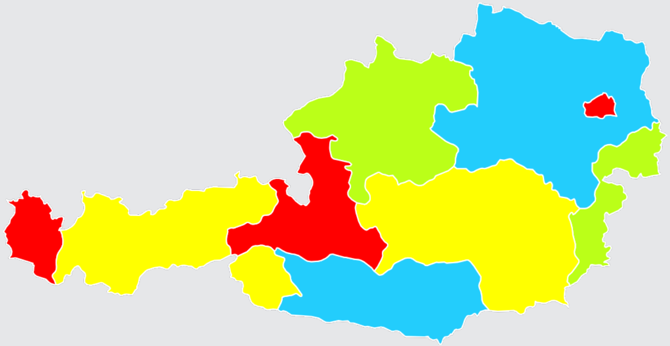
Prolog code

```
:- use_module(library(clpfd)).
nqueens(N,Qs) :-
  length(Qs,N), Qs ins 1 .. N,
  all_different(Qs),
  constraint_queens(Qs), label(Qs).
constraint_queens([]).
  constraint_queens(Qs).
noattack(_,[],_).
noattack(X,[Q|Qs],N) :-
  X #\= Q+N, X #\= Q-N, M is N+1,
  noattack(X,Qs,M).

?- nqueens(8,Xs).
```

logic programming is sometimes taught in elective module

Example (Coloring Austria)



question: do three colors suffice to color Austria?

Example (Programming Task)

function that computes $\sum_{i=1}^{n} i$ (exercise in C programming course at Nagoya University)

some answers:

```
int sum(int x) {
  int i, j, z;
  z = 0;
  for (i = 0; i <= x; i++)
    for (j = 0; j < i; j++)
      z++;
  return z;
}
```

```
int sum(int n) {
  if (n <= 0) {
    return 0;
  } else {
    return (n*(n+1)/2);
  }
}
```

question: are these programs correct?

Greek Alphabet

alpha	α	A	eta	η	H	nu	ν	N	tau	τ	T
beta	β	B	theta	$\theta\ \vartheta$	Θ	xi	ξ	Ξ	upsilon	υ	Υ
gamma	γ	Γ	iota	ι	I	omicron	o	O	phi	$\phi\ \varphi$	Φ
delta	δ	Δ	kappa	κ	K	pi	π	Π	chi	χ	X
epsilon	$\epsilon\ \varepsilon$	E	lambda	λ	Λ	rho	ρ	P	psi	ψ	Ψ
zeta	ζ	Z	mu	μ	M	sigma	$\sigma\ \varsigma$	Σ	omega	ω	Ω

Outline

Part I: Propositional Logic

algebraic normal forms, binary decision diagrams, conjunctive normal forms, DPLL, Horn formulas, natural deduction, Post's adequacy theorem, resolution, SAT, semantics, sorting networks, soundness and completeness, syntax, Tseitin's transformation

Part II: Predicate Logic

natural deduction, quantifier equivalences, resolution, semantics, Skolemization, syntax, undecidability, unification

Part III: Model Checking

adequacy, branching-time temporal logic, CTL*, fairness, linear-time temporal logic, model checking algorithms, symbolic model checking

Outline

Definition

propositional formulas are built from

- atoms $p, q, r, p_1, p_2, \ldots$
- bottom $\bot$
- top $\top$
- negation $\neg$ $\neg p$ "not p"
- conjunction $\wedge$ $p \wedge q$ "p and q"
- disjunction $\vee$ $p \vee q$ "p or q"
- implication $\rightarrow$ $p \rightarrow q$ "if p then q"

according to following Backus–Naur Form:

$$\varphi ::= p \mid \bot \mid \top \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi)$$

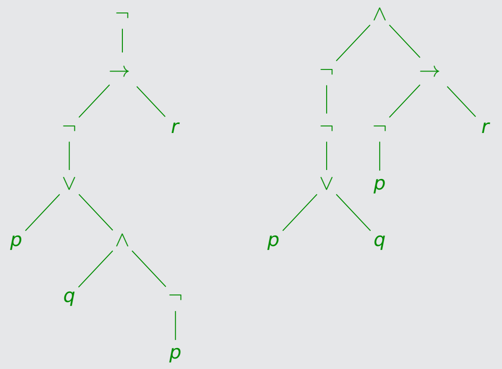
Notational Conventions

- binding precedence $\neg \quad > \quad \wedge, \vee \quad > \quad \rightarrow$
- omit outer parentheses
- $\rightarrow$, $\wedge$, $\vee$ are right-associative: $p \rightarrow q \rightarrow r$ denotes $p \rightarrow (q \rightarrow r)$

Example

formula $\neg(\neg(p \vee (q \wedge \neg p)) \rightarrow r)$ $\neg\neg(p \vee q) \wedge (\neg p \rightarrow r)$ $\neg\neg p \vee q \wedge \neg p \rightarrow r$

parse tree



?

Outline

Definitions (Boole 1854)

- valuation (truth assignment) is mapping $v\colon \{p \mid p \text{ is atom}\} \rightarrow \{\mathsf{T}, \mathsf{F}\}$
- $\bar{v}$ is extension of v to formulas: truth values

$$\bar{v}(p) = v(p) \qquad \bar{v}(\varphi \vee \psi) = \begin{cases} \mathsf{F} & \text{if } \bar{v}(\varphi) = \bar{v}(\psi) = \mathsf{F} \\ \mathsf{T} & \text{otherwise} \end{cases}$$

$$\bar{v}(\bot) = \mathsf{F} \qquad \bar{v}(\top) = \mathsf{T} \qquad \bar{v}(\varphi \wedge \psi) = \begin{cases} \mathsf{T} & \text{if } \bar{v}(\varphi) = \bar{v}(\psi) = \mathsf{T} \\ \mathsf{F} & \text{otherwise} \end{cases}$$

$$\bar{v}(\neg\varphi) = \begin{cases} \mathsf{T} & \text{if } \bar{v}(\varphi) = \mathsf{F} \\ \mathsf{F} & \text{otherwise} \end{cases} \qquad \bar{v}(\varphi \rightarrow \psi) = \begin{cases} \mathsf{F} & \text{if } \bar{v}(\varphi) = \mathsf{T} \text{ and } \bar{v}(\psi) = \mathsf{F} \\ \mathsf{T} & \text{otherwise} \end{cases}$$

Example

$v(p) = \mathsf{T}$ and $v(q) = \mathsf{F} \implies \bar{v}(p \wedge \neg q \rightarrow \neg p) = \mathsf{F}$

Definition

truth tables

φ	$\neg\varphi$
T	F
F	T

φ	ψ	$\varphi \wedge \psi$	$\varphi \vee \psi$	$\varphi \rightarrow \psi$
T	T	T	T	T
T	F	F	T	F
F	T	F	T	T
F	F	F	F	T

truth tables for propositional formulas are constructed bottom-up

Example ❶

p	q	$\neg p$	$\neg q$	$p \to \neg q$	$q \vee \neg p$	$(p \to \neg q) \to (q \vee \neg p)$
T	T	F	F	F	T	T
T	F	F	T	T	F	F
F	T	T	F	T	T	T
F	F	T	T	T	T	T

Example ❷

p	q	$(p$	$\to$	$\neg q)$	$\to$	$(q$	$\vee$	$\neg p)$
T	T				T		T	
T	F		T	T	F		F	F
F	T		T		T		T	
F	F		T		T		T	T

Definitions

- semantic entailment

$$\varphi_1, \varphi_2, \ldots, \varphi_n \vDash \psi$$

  if $\bar{v}(\psi) = \text{T}$ whenever $\bar{v}(\varphi_1) = \bar{v}(\varphi_2) = \cdots = \bar{v}(\varphi_n) = \text{T}$ for every valuation v

- tautology is formula φ such that $\vDash \varphi$

Examples ❶

p	q	p	$\to$	q	$\vDash$	$\neg p$	$\vee$	q
T	T		T				T	
T	F		F					
F	T		T				T	
F	F		T				T	

p	q	p	$\to$	$q,$	p	$\to$	$\neg q$	$\vDash$	$\neg p$
T	T		T			F			
T	F		F						
F	T		T			T			T
F	F		T			T			T

Examples ❷

p	q	p	$\to$	q	$\nvDash$	q	$\to$	p
T	T		T				T	
T	F		F					
F	T		T				F	
F	F		T					

p	q	p	$\to$	$q,$	p	$\to$	$\neg q$	$\nvDash$	q
T	T		T			F			
T	F		F						
F	T		T			T			T
F	F		T			T			F

p	q	p	$\to$	$q,$	p	$\wedge$	$\neg q$	$\vDash$	$\bot$
T	T		T			F			
T	F		F						
F	T		T			F			
F	F		T			F			

Question

$$\vDash \begin{array}{l}(\neg p \wedge \neg q) \vee (s \wedge u) \vee (r \wedge w) \vee (\neg t \wedge \neg u) \vee (p \wedge r) \vee (q \wedge s) \\ \vee (p \wedge t) \vee (q \wedge u) \vee (\neg r \wedge \neg s) \vee (t \wedge v) \vee (\neg v \wedge \neg w)\end{array} \; ?$$

... truth table has $2^8 = 256$ rows ...

	6		1		4		5	
		8	3		5	6		
2								1
8			4		7			6
		6				3		
7			9		1			4
5								2
		7	2		6	9		
	4		5		8		7	

... truth table has $2^{459} > 2^{4 \times 100} = 16^{100} > 10^{100}$ rows ...

Outline

Definitions

formula φ is

- valid if $\bar{v}(\varphi) = \mathsf{T}$ for every valuation v
- satisfiable if $\bar{v}(\varphi) = \mathsf{T}$ for some valuation v

Theorem

formula φ is valid $\iff$ $\neg\varphi$ is unsatisfiable

Proof

$$\begin{aligned} \varphi \text{ is valid} &\iff \bar{v}(\varphi) = \mathsf{T} \quad \text{for every valuation } v \\ &\iff \bar{v}(\neg\varphi) = \mathsf{F} \quad \text{for every valuation } v \\ &\iff \bar{v}(\neg\varphi) = \mathsf{T} \quad \text{for no valuation } v \\ &\iff \neg\varphi \text{ is not satisfiable} \iff \neg\varphi \text{ is unsatisfiable} \end{aligned}$$

Definition

formulas φ and ψ are semantically equivalent ($\varphi \equiv \psi$) if both $\varphi \vDash \psi$ and $\psi \vDash \varphi$

Examples

$$\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi \qquad \neg\neg\varphi \equiv \varphi \qquad \varphi \vee (\psi \wedge \chi) \equiv (\varphi \vee \psi) \wedge (\varphi \vee \chi)$$

Theorem

validity and satisfiability are decidable

Proof

construct truth table of φ and inspect last column:

- φ is valid if and only if all entries are T
- φ is satisfiable if and only if T entry exists

Outline

Particify with session ID **0992 9580**

Question

Given that one and only one answer is correct, which of the following is true ?

- **A** All of the below.
- **B** None of the below.
- **C** One of the above.
- **D** All of the above.
- **E** None of the above.
- **F** None of the above.

Outline

Definitions

- literal is atom p or negation $\neg p$ of atom
- clause is disjunction $\ell_1 \vee \cdots \vee \ell_n$ of literals
- conjunctive normal form (CNF) is conjunction $C_1 \wedge \cdots \wedge C_n$ of clauses
- literals ℓ_1 and ℓ_2 are complementary if $\ell_1 = \neg \ell_2$ or $\neg \ell_1 = \ell_2$

Theorem

validity of CNFs is efficiently decidable:

CNF φ is valid $\iff$ every clause of φ contains complementary literals

Examples

1. CNF

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

complementary literals

not valid witness: $v(p) = v(q) = \mathsf{F}$ and $v(r) = \mathsf{T}$

2. CNF

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

valid

Special Cases

- $\bot$ represents empty clause (no literals)
- $\top$ represents empty CNF (no clauses)

Theorem

for every formula φ there exists CNF ψ such that $\varphi \equiv \psi$

Procedure

① eliminate implications

② push negations towards atoms and remove double negations

③ distribute disjunction over conjunction

$$
\begin{array}{rcl@{\qquad\qquad}rcl}
\varphi \to \psi & \overset{①}{\to} & \neg\varphi \vee \psi & \neg\neg\varphi & \overset{②}{\to} & \varphi \\
\neg(\varphi \wedge \psi) & \overset{②}{\to} & \neg\varphi \vee \neg\psi & \varphi \vee (\psi \wedge \chi) & \overset{③}{\to} & (\varphi \vee \psi) \wedge (\varphi \vee \chi) \\
\neg(\varphi \vee \psi) & \overset{②}{\to} & \neg\varphi \wedge \neg\psi & (\varphi \wedge \psi) \vee \chi & \overset{③}{\to} & (\varphi \vee \chi) \wedge (\psi \vee \chi)
\end{array}
$$

Remark

CNF ψ for formula φ might be exponentially larger

Example (CNFs are not unique)

$$
\begin{array}{l}
\varphi = \neg(p \vee (q \wedge r)) \\
\quad \overset{②}{\to} \ \neg p \wedge \neg(q \wedge r) \ \overset{②}{\to} \ \neg p \wedge (\neg q \vee \neg r)
\end{array}
$$

$$
\begin{array}{l}
\varphi = \neg(p \vee (q \wedge r)) \\
\quad \overset{③}{\to} \ \neg((p \vee q) \wedge (p \vee r)) \ \overset{②}{\to} \ \neg(p \vee q) \vee \neg(p \vee r) \\
\quad \overset{②}{\to} \ (\neg p \wedge \neg q) \vee \neg(p \vee r) \ \overset{②}{\to} \ (\neg p \wedge \neg q) \vee (\neg p \wedge \neg r) \\
\quad \overset{③}{\to} \ ((\neg p \wedge \neg q) \vee \neg p) \wedge ((\neg p \wedge \neg q) \vee \neg r) \\
\quad \overset{③}{\to} \ ((\neg p \vee \neg p) \wedge (\neg q \vee \neg p)) \wedge ((\neg p \wedge \neg q) \vee \neg r) \\
\quad \overset{③}{\to} \ ((\neg p \vee \neg p) \wedge (\neg q \vee \neg p)) \wedge ((\neg p \vee \neg r) \wedge (\neg q \vee \neg r))
\end{array}
$$

CNFs are not unique, even if rules ①, ②, ③ are applied in order

Procedure (extended)

⓪ simplify formulas with $\bot$ and $\top$

① eliminate implications

② push negations towards atoms and remove double negations

③ distribute disjunction over conjunction

$$
\begin{array}{rcl@{\qquad}rcl@{\qquad}rcl@{\qquad}rcl}
\neg\bot & \overset{⓪}{\to} & \top & \bot \wedge \varphi & \overset{⓪}{\to} & \bot & \bot \vee \varphi & \overset{⓪}{\to} & \varphi & \bot \to \varphi & \overset{⓪}{\to} & \top \\
\neg\top & \overset{⓪}{\to} & \bot & \top \wedge \varphi & \overset{⓪}{\to} & \varphi & \top \vee \varphi & \overset{⓪}{\to} & \top & \top \to \varphi & \overset{⓪}{\to} & \varphi \\
 & & & \varphi \wedge \bot & \overset{⓪}{\to} & \bot & \varphi \vee \bot & \overset{⓪}{\to} & \varphi & \varphi \to \bot & \overset{⓪}{\to} & \neg\varphi \\
 & & & \varphi \wedge \top & \overset{⓪}{\to} & \varphi & \varphi \vee \top & \overset{⓪}{\to} & \top & \varphi \to \top & \overset{⓪}{\to} & \top
\end{array}
$$

Example

$$
\begin{array}{l}
p \vee (q \wedge (\top \to (\neg p \vee \bot)) \to (\top \wedge \neg q)) \\
\quad \overset{⓪}{\to} \ p \vee (q \wedge (\top \to \neg p) \to (\top \wedge \neg q)) \ \overset{⓪}{\to} \ p \vee (q \wedge (\top \to \neg p) \to \neg q) \\
\quad \overset{⓪}{\to} \ p \vee (q \wedge \neg p \to \neg q) \ \overset{①}{\to} \ p \vee (\neg(q \wedge \neg p) \vee \neg q) \\
\quad \overset{②}{\to} \ p \vee ((\neg q \vee \neg\neg p) \vee \neg q) \ \overset{②}{\to} \ p \vee ((\neg q \vee p) \vee \neg q)
\end{array}
$$

Example (CNF from truth table)

$\varphi = \neg(p \vee (q \wedge r))$

p	q	r	φ	p	q	r	φ
T	T	T	F	F	T	T	F
T	T	F	F	F	T	F	T
T	F	T	F	F	F	T	T
T	F	F	F	F	F	F	T

$$\begin{aligned}\varphi &\equiv \neg((p \wedge q \wedge r) \vee (p \wedge q \wedge \neg r) \vee (p \wedge \neg q \wedge r) \vee (p \wedge \neg q \wedge \neg r) \vee (\neg p \wedge q \wedge r))\\ &\equiv (\neg p \vee \neg q \vee \neg r) \wedge (\neg p \vee \neg q \vee r) \wedge (\neg p \vee q \vee \neg r) \wedge (\neg p \vee q \vee r) \wedge (p \vee \neg q \vee \neg r)\end{aligned}$$

Theorem

for every formula φ there exists CNF ψ such that $\varphi \equiv \psi$

Deterministic Procedure

① eliminate implications

function IMPL_FREE(φ):
begin function
case φ is atom: **return** φ
φ is $\neg\varphi_1$: **return** $\neg$ IMPL_FREE(φ_1)
φ is $\varphi_1 \wedge \varphi_2$: **return** IMPL_FREE($\varphi_1$) $\wedge$ IMPL_FREE(φ_2)
φ is $\varphi_1 \vee \varphi_2$: **return** IMPL_FREE(φ_1) $\vee$ IMPL_FREE(φ_2)
φ is $\varphi_1 \rightarrow \varphi_2$: **return** $\neg$ IMPL_FREE(φ_1) $\vee$ IMPL_FREE(φ_2)
end case
end function

Theorem

for every formula φ there exists CNF ψ such that $\varphi \equiv \psi$

Deterministic Procedure

② push negations towards atoms and remove double negations

function NNF(φ):
begin function
case φ is literal: **return** φ
φ is $\neg\neg\varphi_1$: **return** NNF(φ_1)
φ is $\varphi_1 \wedge \varphi_2$: **return** NNF($\varphi_1$) $\wedge$ NNF(φ_2)
φ is $\varphi_1 \vee \varphi_2$: **return** NNF(φ_1) $\vee$ NNF(φ_2)
φ is $\neg(\varphi_1 \wedge \varphi_2)$: **return** NNF($\neg\varphi_1$) $\vee$ NNF($\neg\varphi_2$)
φ is $\neg(\varphi_1 \vee \varphi_2)$: **return** NNF($\neg\varphi_1$) $\wedge$ NNF($\neg\varphi_2$)
end case
end function

Theorem

for every formula φ there exists CNF ψ such that $\varphi \equiv \psi$

Deterministic Procedure

③ distribute disjunction over conjunction

function CNF(φ):
begin function
case φ is literal: **return** φ
φ is $\varphi_1 \wedge \varphi_2$: **return** CNF($\varphi_1$) $\wedge$ CNF(φ_2)
φ is $\varphi_1 \vee \varphi_2$: **return** DISTR(CNF(φ_1), CNF(φ_2))
end case
end function

Theorem

for every formula φ there exists CNF ψ such that $\varphi \equiv \psi$

Deterministic Procedure

③ distribute disjunction over conjunction

function $\text{DISTR}(\eta_1, \eta_2)$:
begin function
case η_1 is $\eta_{11} \wedge \eta_{12}$: **return** $\text{DISTR}(\eta_{11}, \eta_2) \wedge \text{DISTR}(\eta_{12}, \eta_2)$
η_2 is $\eta_{21} \wedge \eta_{22}$: **return** $\text{DISTR}(\eta_1, \eta_{21}) \wedge \text{DISTR}(\eta_1, \eta_{22})$
otherwise: **return** $\eta_1 \vee \eta_2$
end case
end function

Theorem

for every formula φ there exists CNF ψ such that $\varphi \equiv \psi$

Deterministic Procedure

① eliminate implications

② push negations towards atoms and remove double negations

③ distribute disjunction over conjunction

Theorem

❶ $\text{CNF}(\text{NNF}(\text{IMPL\_FREE}(\varphi)))$ is CNF

❷ $\text{CNF}(\text{NNF}(\text{IMPL\_FREE}(\varphi))) \equiv \varphi$

❸ executing $\text{CNF}(\text{NNF}(\text{IMPL\_FREE}(\varphi)))$ terminates

Outline

Huth and Ryan

- Section 1.1
- Section 1.3
- Sections 1.4.1 and 1.4.2
- Sections 1.5.1 and 1.5.2

Differences (slides – book)

- role of $\bot$ and $\top$
- terminology concerning CNFs

Important Concepts

- atom
- bottom
- clause
- complementary literals
- conjunction
- conjunctive normal form
- disjunction
- disjunctive normal form
- implication
- literal
- negation
- top
- right-associativity
- satisfiability
- semantic entailment
- semantic equivalence
- tautology
- truth table
- truth values
- validity
- valuation

homework for March 7 (this week)