EDAA40
Discrete Structures in Computer Science

7: Propositional logic

Jörn W. Janneck, Dept. of Computer Science, Lund University

mechanizing belief management

Gottfried Wilhelm von Leibniz
1646-1716

A major goal of any logic is to *mechanize reasoning*:
we figure out *truth* through manipulating strings of symbols according to some rules.
Ultimately, this can be done by a machine.

We need:
1. a way to represent the truth/falsehood of propositions
2. a system of symbols for constructing, combining, relating propositions
3. rules for manipulating and reasoning about them

We assume that all propositions are either true or false (*bivalence*).
Truth and falsehood are represented by 1 and 0, respectively.

basic logic connectives

α	$\neg\alpha$
1	0
0	1

α	β	$\alpha \wedge \beta$	$\alpha \vee \beta$	$\alpha \rightarrow \beta$	$\alpha \leftrightarrow \beta$	$\alpha \veebar \beta$	$\alpha \bar{\wedge} \beta$
1	1	1	1	1	1	0	0
1	0	0	1	0	0	1	1
0	1	0	1	1	0	1	1
0	0	0	0	1	1	0	1
		and	or	implies	iff	xor	nand

The "and" connective is also called *conjunction*, the "or" connective *disjunction*.

from truth table to formula

p	q	r	f(p, q, r)	
1	1	1	1	$\leftarrow$ $p \wedge q \wedge r$
1	1	0	0	
1	0	1	1	$\leftarrow$ $p \wedge \neg q \wedge r$
1	0	0	0	
0	1	1	0	
0	1	0	0	
0	0	1	1	$\leftarrow$ $\neg p \wedge \neg q \wedge r$
0	0	0	0	

Can this expression be simplified?

Are there truth tables for which this method does not produce a formula?

What does this mean for the connectives not, and, or?

$f : (p, q, r) \mapsto (p \wedge q \wedge r) \vee (p \wedge \neg q \wedge r) \vee (\neg p \wedge \neg q \wedge r)$

relating connectives

α	β	$\alpha \wedge \beta$	$\alpha \vee \beta$	$\alpha \rightarrow \beta$	$\alpha \leftrightarrow \beta$	$\alpha \veebar \beta$	$\alpha \barwedge \beta$
1	1	1	1	1	1	0	0
1	0	0	1	0	0	1	1
0	1	0	1	1	0	1	1
0	0	0	0	1	1	0	1
		and	or	implies	iff	xor	nand

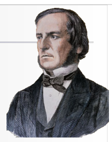
Express *or* in terms of *not* and *and*. $\alpha \vee \beta = \ldots$

Is there one connective to express them all?

$\neg\alpha = \alpha \barwedge \alpha$

$\alpha \wedge \beta = \neg(\alpha \barwedge \beta) = (\alpha \barwedge \beta) \barwedge (\alpha \barwedge \beta)$

Boolean algebra(s)



George Boole
1815-1864

sidebar

Propositional logic is a Boolean algebra.

A *Boolean algebra* is a set B, with one unary and two binary operations that have the following properties, for all $x, y, z \in B$:

there is a $z \in B : x \vee z = x$ and $x \wedge \neg x = z$

there is a $e \in B : x \wedge e = x$ and $x \vee \neg x = e$

$x \vee y = y \vee x$ and $x \wedge y = y \wedge x$

$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$

$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$

Propositional logic is a Boolean algebra with B$=\{1, 0\}$, $z = 0$, e $= 1$, and the operations and ($\wedge$), or ($\vee$), not ($\neg$).

Can you think of other Boolean algebras?

language layers in logic

In logic, we are concerned with several layers of language, and the truth of statements in them.

1. Elementary propositions.
   These are treated as "atomic", all we care about is that they are either true or false.
   In logic, we represent them with *elementary letters*.

   "All birds can fly."
   "17 is a prime number."
   "4 > 11"
   $p, q, r, \ldots$

2. Formulae.
   Expressions constructed recursively from elementary letters and *connectives*.
   We represent them with greek symbols.

   $(p \wedge q) \rightarrow (s \vee t)$
   $\neg p \vee q$
   $\alpha, \beta, \gamma, \ldots$

3. Relations between formulae.
   This is what we use to reason about formulae.

   $\alpha_1, \ldots, \alpha_n \vdash \beta$
   $\alpha \dashv\vdash \beta$

assignments & valuations

Given a set of elementary letters E, an *assignment* is a function

$$v : E \longrightarrow \{1, 0\}$$

E

p	q	$p \wedge (p \rightarrow q)$
1	1	1
1	0	0
0	1	0
0	0	0

v

assignments & valuations

Given a set of elementary letters E and an assignment v, a *valuation* is the recursive extension of v over the set F[E] of formulae in E generated by a set of rules F:

$$v^+ : F[E] \longrightarrow \{1, 0\}$$

F[E]

p	q	$p \wedge (p \rightarrow q)$
1	1	1
1	0	0
0	1	0
0	0	0

Example:

$F = \{F_1, F_2, F_3\}$
$F_1 = \{(\alpha, \neg\alpha) : \alpha \in U\}$
$F_2 = \{(\alpha, \beta, (\alpha \wedge \beta)) : \alpha, \beta \in U\}$
$F_3 = \{(\alpha, \beta, (\alpha \vee \beta)) : \alpha, \beta \in U\}$

$$v^+ : \gamma \mapsto \begin{cases} v(\gamma) & \text{for } \gamma \in E \\ f_\neg(v^+(\alpha)) & \text{for } \gamma = \neg\alpha \\ f_\wedge(v^+(\alpha), v^+(\beta)) & \text{for } \gamma = (\alpha \wedge \beta), \alpha, \beta \in F[E] \\ f_\vee(v^+(\alpha), v^+(\beta)) & \text{for } \gamma = (\alpha \vee \beta), \alpha, \beta \in F[E] \end{cases}$$

⚠ From here on out, we will use v for both an assignment and its valuation.

tautological implication

Given a set of formulae A and a formula β we say that A *tautologically implies* β if there is no valuation v such that

$$v(\alpha) = 1 \text{ for all } \alpha \in A \text{ and } v(\beta) = 0$$

$A \vdash \beta$

$A \models \beta$

Example:
(modus ponens) $p, p \to q \vdash q$

p	q	$p \to q$
1	1	1
1	0	0
0	1	1
0	0	1

Name	LHS	RHS
Simplification, $\wedge^-$	$\alpha\wedge\beta$	α
	$\alpha\wedge\beta$	β
Conjunction, $\wedge^+$	α, β	$\alpha\wedge\beta$
Disjunction, $\vee^+$	α	$\alpha\vee\beta$
	β	$\alpha\vee\beta$
Modus ponens, MP, $\to^-$	$\alpha, \alpha\to\beta$	β
Modus tollens, MT	$\neg\beta, \alpha\to\beta$	$\neg\alpha$
Disjunctive syllogism, DS	$\alpha\vee\beta, \neg\alpha$	β
Transitivity	$\alpha\to\beta, \beta\to\gamma$	$\alpha\to\gamma$
Material implication	β	$\alpha\to\beta$
	$\neg\alpha$	$\alpha\to\beta$
Limiting cases	γ	$\beta\vee\neg\beta$
	$\alpha\wedge\neg\alpha$	γ

SLAM, p. 196

tautological equivalence

Given two formulae α and β, we say that they are *tautologically equivalent* if they tautologically imply each other.

$\alpha \dashv\vdash \beta$

Name	LHS	RHS
Double negation	α	$\neg\neg\alpha$
Commutation for $\wedge$	$\alpha\wedge\beta$	$\beta\wedge\alpha$
Association for $\wedge$	$\alpha\wedge(\beta\wedge\gamma)$	$(\alpha\wedge\beta)\wedge\gamma$
Commutation for $\vee$	$\alpha\vee\beta$	$\beta\vee\alpha$
Association for $\vee$	$\alpha\vee(\beta\vee\gamma)$	$(\alpha\vee\beta)\vee\gamma$
Distribution of $\wedge$ over $\vee$	$\alpha\wedge(\beta\vee\gamma)$	$(\alpha\wedge\beta)\vee(\alpha\wedge\gamma)$
Distribution of $\vee$ over $\wedge$	$\alpha\vee(\beta\wedge\gamma)$	$(\alpha\vee\beta)\wedge(\alpha\vee\gamma)$
Absorption	α	$\alpha\wedge(\alpha\vee\beta)$
	α	$\alpha\vee(\alpha\wedge\beta)$
Expansion	α	$(\alpha\wedge\beta)\vee(\alpha\wedge\neg\beta)$
	α	$(\alpha\vee\beta)\wedge(\alpha\vee\neg\beta)$
de Morgan	$\neg(\alpha\wedge\beta)$	$\neg\alpha\vee\neg\beta$
	$\neg(\alpha\vee\beta)$	$\neg\alpha\wedge\neg\beta$
	$\alpha\wedge\beta$	$\neg(\neg\alpha\vee\neg\beta)$
	$\alpha\vee\beta$	$\neg(\neg\alpha\wedge\neg\beta)$
Limiting cases	$\alpha\wedge\neg\alpha$	$\beta\wedge\neg\beta$
	$\alpha\vee\neg\alpha$	$\beta\vee\neg\beta$

SLAM, p. 198

tautological equivalence

Name	LHS	RHS
Contraposition	$\alpha\to\beta$	$\neg\beta\to\neg\alpha$
	$\alpha\to\neg\beta$	$\beta\to\neg\alpha$
	$\neg\alpha\to\beta$	$\neg\beta\to\alpha$
Import/export	$\alpha\to(\beta\to\gamma)$	$(\alpha\wedge\beta)\to\gamma$
	$\alpha\to(\beta\to\gamma)$	$\beta\to(\alpha\to\gamma)$
Consequential mirabilis (miraculous consequence)	$\alpha\to\neg\alpha$	$\neg\alpha$
	$\neg\alpha\to\alpha$	α
Commutation for $\leftrightarrow$	$\alpha\leftrightarrow\beta$	$\beta\leftrightarrow\alpha$
Association for $\leftrightarrow$	$\alpha\leftrightarrow(\beta\leftrightarrow\gamma)$	$(\alpha\leftrightarrow\beta)\leftrightarrow\gamma$
$\neg$ through $\leftrightarrow$	$\neg(\alpha\leftrightarrow\beta)$	$\alpha\leftrightarrow\neg\beta$
Translations between connectives	$\alpha\leftrightarrow\beta$	$(\alpha\to\beta)\wedge(\beta\to\alpha)$
	$\alpha\leftrightarrow\beta$	$(\alpha\wedge\beta)\vee(\neg\alpha\wedge\neg\beta)$
	$\alpha\to\beta$	$\neg(\alpha\wedge\neg\beta)$
	$\alpha\to\beta$	$\neg\alpha\vee\beta$
	$\alpha\vee\beta$	$\neg\alpha\to\beta$
Translations of negations of connectives	$\neg(\alpha\to\beta)$	$\alpha\wedge\neg\beta$
	$\neg(\alpha\wedge\beta)$	$\alpha\to\neg\beta$
	$\neg(\alpha\leftrightarrow\beta)$	$(\alpha\wedge\neg\beta)\vee(\beta\wedge\neg\alpha)$

SLAM, p. 199

tautology and contradiction

A formula α ...
... is a *tautology* iff $v(\alpha) = 1$ for every valuation v
... is a *contradiction* iff $v(\alpha) = 0$ for every valuation v
... *contingent* otherwise.

DNF: disjunctive normal form

A *normal form* is a transformation of a formula into another equivalent form that has particular properties. Some definitions:

A *literal* is an elementary letter or its negation.
A *basic conjunction* is a conjunction of literals, without repetition.

Examples: $p \wedge q \wedge \neg r$ $\quad p \quad$ $\neg p \wedge \neg q$

A formula is in *disjunctive normal form* if it is a disjunction of basic conjunctions.

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

A formula is in *full disjunctive normal form* every letter occurs in every basic conjunction.

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

constructing the DNF

Given a formula, there are two common ways of constructing its DNF:

1. via its truth table
2. through successive syntactic transformations

Basic algorithm for building a DNF (SLAM p. 204):

1. express all connectives through negation, conjunction, disjunction
2. move negations inward (de Morgan), eliminate double negation
3. move conjunctions inward (distribution)
4. remove repetitions in basic conjunctions
5. remove basic conjunctions with a letter and its negation

Example: $\neg((p \vee \neg q) \rightarrow (p \wedge \neg q))$

1. $\neg(\neg(p \vee \neg q) \vee (p \wedge \neg q))$

2.1 $\neg\neg(p \vee \neg q) \wedge \neg(p \wedge \neg q)$

2.2 $\neg\neg(p \vee \neg q) \wedge (\neg p \vee q)$

2.3 $(p \vee \neg q) \wedge (\neg p \vee q)$

3. $(p \wedge \neg p) \vee (p \wedge q) \vee (\neg q \wedge \neg p) \vee (\neg q \wedge q)$

5. $(p \wedge q) \vee (\neg p \wedge \neg q)$