## EDAA40 **Discrete Structures in Computer Science**

## 8: Propositional logic

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

## mechanizing belief management

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:

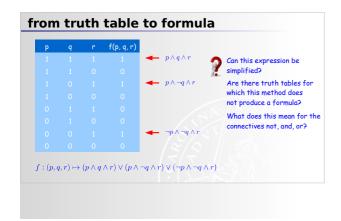
  1. a way to represent the truth/falsehood of propositions

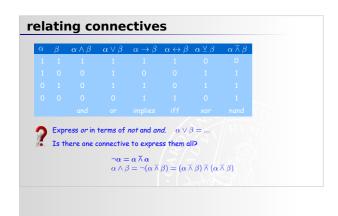
  2. a system of symbols for constructing, combining, relating propositions

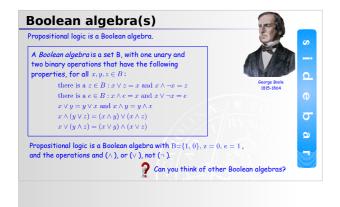
  3. rules for manipulating and reasoning about them

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

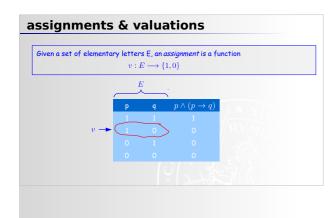
# basic logic connectives $\neg \alpha \qquad \qquad \alpha \qquad \beta \qquad \alpha \wedge \beta \qquad \alpha \vee \beta \qquad \alpha \to \beta \quad \alpha \leftrightarrow \beta \quad \alpha \veebar \beta \qquad \alpha \ \overline{\wedge} \ \beta$ The "and" connective is also called *conjunction*, the "or" connective *disjunction*.

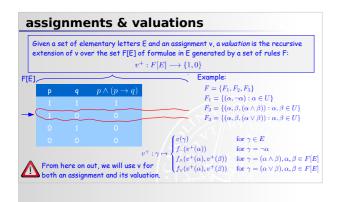


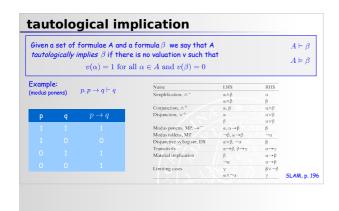




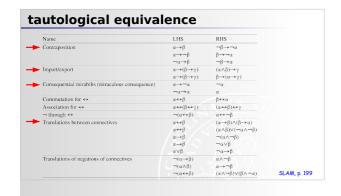
In logic, we are concerned with several layers of language, and the truth of statements in them.				
Elementary propositions.     These are treated as "atomic", all we care about is that they are either true or false.     In logic, we represent them with elementary letters.	"All birds can fly." "17 is a prime number. "4 > 11" $p, q, r, \dots$			
Formulae.     Expressions constructed recursively from elementary letters and <i>connectives</i> .     We represent them with greek symbols.	$ \begin{array}{l} (p \wedge q) \rightarrow (s \vee t) \\ \neg p \vee q \\ \alpha, \beta, \gamma, \dots \end{array} $			
3. Relations between formulae. This is what we use to reason about formulae.	$\alpha_1,, \alpha_n \vdash \beta$ $\alpha \dashv \vdash \beta$			







Given two formulae $lpha$ and $eta$ , we say that they are <i>tautologically equivalent</i> $\alpha+\beta$ f they tautologically imply each other.					
,					
	Name	LHS	RHS		
	Double negation	a	$\neg \neg \alpha$		
	Commutation for ∧	αΛβ	βΛα		
	Association for ∧	αΛ(βΛγ)	(αΛβ)Λγ		
	Commutation for ∨	ανβ	βVα		
	Association for ∨	αν(βνγ)	(ανβ)νγ		
	Distribution of ∧ over ∨	α∧(β∨γ)	$(\alpha \wedge \beta) \vee (\alpha \wedge \gamma)$		
	Distribution of ∨ over ∧	αν(βΛγ)	(αVβ)Λ(αVγ)		
	Absorption	α	αΛ(αVβ) αV(αΛβ)		
	Expansion	a a	(a∧6)∨(a∧¬8)		
	Expansion	, 0	(a∨8)∧(a∨¬8)		
	de Morgan	¬(a∧8)	7x778		
	ac magain	-(α v β)	¬v∧¬8		
		αΛβ	$\neg(\neg\alpha \lor \neg\beta)$		
		ανβ	$\neg(\neg\alpha \land \neg\beta)$		
	Limiting cases	a∧¬a	β∧¬β	SLAM, p. 19	
		α∨⊸α	$\beta \vee \neg \beta$		



## tautology and contradiction **A formula** $\alpha$ ... ... 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** Anormal 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. 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. $\textbf{Example:} \quad (p \wedge q \wedge \neg r) \vee (\neg p \wedge \neg q \wedge r) \vee (\neg p \wedge \neg q \wedge \neg r)$

## constructing the DNF

Given a formula, there are two common ways of constructing its DNF: 1. via its truth table

- ${\bf 2.\ through\ successive\ syntactic\ transformations}$

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

- sasic algorithm for building a DINT (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

- $\begin{array}{ll} \textbf{Example:} & \neg((p \vee \neg q) \rightarrow (p \wedge \neg q)) \\ \textbf{1.} & \neg(\neg(p \vee \neg q) \vee (p \wedge \neg q)) & \textbf{2.3} \ (p \vee \neg q) \wedge (\neg p \vee q) \end{array}$ 
  - $\textbf{2.1} \ \neg \neg (p \vee \neg q) \wedge \neg (p \wedge \neg q) \quad \textbf{3.} \quad (p \wedge \neg p) \vee (p \wedge q) \vee (\neg q \wedge \neg p) \vee (\neg q \wedge q)$
  - **2.2**  $\neg\neg(p \lor \neg q) \land (\neg p \lor q)$  **5.**  $(p \land q) \lor (\neg p \land \neg q)$