PROPOSITIONAL LOGIC: MORE ON NORMAL FORMS

VL Logic, Lecture 3, WS 20/21

Armin Biere, Martina Seidl

Institute for Formal Models and Verification
Johannes Kepler University Linz

JOHANNES KEPLER
UNIVERSITY LINZ

PROPOSITIONAL FORMULAS: SATISFIABILITY EQUIVALENCES

VL Logic

Part I: Propositional Logic

JOHANNES KEPLER
UNIVERSITY LINZ

Recap: Semantic Equivalence

Two formulas ϕ and ψ are **semanticly equivalent** (written as $\phi \Leftrightarrow \psi$) iff forall interpretations $[.]_\nu$ it holds that $[\phi]_\nu = [\psi]_\nu$.

- $\Leftrightarrow$ is a **meta-symbol**, i.e., it is not part of the language.
- $\phi \Leftrightarrow \psi$ iff $\phi \leftrightarrow \psi$ is valid, i.e., we can express semantics by means of syntactics.
- If ϕ and ψ are not equivalent, we write $\phi \not\Leftrightarrow \psi$.

Example:

- $a \vee \neg a \not\Leftrightarrow b \rightarrow \neg b$
- $a \vee \neg a \Leftrightarrow b \vee \neg b$
- $(a \vee b) \wedge \neg(a \vee b) \Leftrightarrow \perp$
- $a \leftrightarrow (b \leftrightarrow c) \Leftrightarrow (a \leftrightarrow b) \leftrightarrow c$

Satisfiability Equivalence

Two formulas ϕ and ψ are **satisfiability-equivalent** (written as $\phi \Leftrightarrow_{SAT} \psi$) iff both formulas are satisfiable or both are contradictory.

- Satisfiability-equivalent formulas are not necessarily satisfied by the same assignments.
- Satisfiability equivalence is a weaker property than semantic equivalence.
- Often sufficient for simplification rules: If the complicated formula is satisfiable then also the simplified formula is satisfiable.

Example: Satisfiability Equivalence

positive pure literal elimination rule:

If a literal x occurs in an NNF formula but $\neg x$ does not occur in this formula, then x can be substituted by $\top$. The resulting formula is satisfiability-equivalent.

Example:

- $x \Leftrightarrow_{SAT} \top$, but $x \not\Leftrightarrow \top$
- $(a \wedge b) \vee (\neg c \wedge a) \Leftrightarrow_{SAT} b \vee \neg c$, but $(a \wedge b) \vee (\neg c \wedge a) \not\Leftrightarrow b \vee \neg c$

PROPOSITIONAL FORMULAS: FUNCTIONAL COMPLETENESS

VL Logic

Part I: Propositional Logic

JOHANNES KEPLER
UNIVERSITY LINZ

Examples of Semantic Equivalences (1/2)

$\phi \wedge \psi \Leftrightarrow \psi \wedge \phi$	$\phi \vee \psi \Leftrightarrow \psi \vee \phi$	commutativity
$\phi \wedge (\psi \wedge \gamma) \Leftrightarrow (\phi \wedge \psi) \wedge \gamma$	$\phi \vee (\psi \vee \gamma) \Leftrightarrow (\phi \vee \psi) \vee \gamma$	associativity
$\phi \wedge (\phi \vee \psi) \Leftrightarrow \phi$	$\phi \vee (\phi \wedge \psi) \Leftrightarrow \phi$	absorption
$\phi \wedge (\psi \vee \gamma) \Leftrightarrow (\phi \wedge \psi) \vee (\phi \wedge \gamma)$	$\phi \vee (\psi \wedge \gamma) \Leftrightarrow (\phi \vee \psi) \wedge (\phi \vee \gamma)$	distributivity
$\neg(\phi \wedge \psi) \Leftrightarrow \neg\phi \vee \neg\psi$	$\neg(\phi \vee \psi) \Leftrightarrow \neg\phi \wedge \neg\psi$	laws of De Morgan
$\phi \leftrightarrow \psi \Leftrightarrow (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$	$\phi \leftrightarrow \psi \Leftrightarrow (\phi \wedge \psi) \vee (\neg\phi \wedge \neg\psi)$	synt. equivalence

Examples of Semantic Equivalences (2/2)

$\phi \lor \psi \Leftrightarrow \neg\phi \to \psi$	$\phi \to \psi \Leftrightarrow \neg\psi \to \neg\phi$	implications
$\phi \land \neg\phi \Leftrightarrow \bot$	$\phi \lor \neg\phi \Leftrightarrow \top$	complement
$\neg\neg\phi \Leftrightarrow \phi$		double negation
$\phi \land \top \Leftrightarrow \phi$	$\phi \lor \bot \Leftrightarrow \phi$	neutrality
$\phi \lor \top \Leftrightarrow \top$	$\phi \land \bot \Leftrightarrow \bot$	
$\neg\top \Leftrightarrow \bot$	$\neg\bot \Leftrightarrow \top$	

Functional Completeness

- In propositional logic there are
  - 2 functions of arity 0 ($\top, \bot$)
  - 4 functions of arity 1 (e.g., not)
  - 16 functions of arity 2 (e.g., and, or, ...)
  - 2^{2^n} functions of arity n.
- A function of arity n has 2^n different combinations of arguments (lines in the truth table).
- A functions maps its arguments either to **1** or **0**.

A set of functions is called **functional complete** for propositional logic iff it is possible to express all other functions of propositional logic with functions from this set.

$\{\neg, \wedge\}$, $\{\neg, \vee\}$, {nand} are functional complete.

PROPOSITIONAL FORMULAS: NORMAL FORMS

VL Logic

Part I: Propositional Logic

JOHANNES KEPLER
UNIVERSITY LINZ

Negation Normal Form (1/2)

Negation Normal Form (NNF) is defined as follows:

- Literals and truth constants are in NNF;
- $\phi \circ \psi$ ($\circ \in \{\vee, \wedge\}$) is in NNF iff ϕ and ψ are in NNF;
- no other formulas are in NNF.

In other words: A formula in NNF contains only conjunctions, disjunctions, and negations and negations only occur in front of variables and constants.

Negation Normal Form (2/2)

If a formula is in negation normal form then

- in the formula tree, nodes with negation symbols only occur directly before leaves.
- there are no subformulas of the form $\neg\phi$ where ϕ is something else than a variable or a constant.
- it does not contain NAND, NOR, XOR, equivalence, and implication connectives.

Example: The formula $((x \vee \neg x_1) \wedge (x \vee (\neg z \vee \neg x_1)))$ is in NNF but $\neg((x \vee \neg x_1) \wedge (x \vee (\neg z \vee \neg x_1)))$ is not in NNF.

Conjunctive Normal Form (CNF)

A propositional formula is in **conjunctive normal form** (CNF) iff it is a conjunction of clauses.

A formula in conjunctive normal form is

- in negation normal form
- $\top$ if it contains no clauses
- easy to check whether it can be refuted

remark: CNF is the input of most SAT-solvers (DIMACS format)

Disjunctive Normal Form (DNF)

A propositional formula is in **disjunctive normal form (DNF)** if it is a disjunction of cubes.

A formula in disjunctive normal form is

- in negation normal form
- $\perp$ if it contains no cubes
- easy to check whether it can be satisfied

Examples for CNF and DNF

Examples CNF

- $\top$
- $\bot$
- a
- $\neg a$
- $l_1 \wedge l_2 \wedge l_3$
- $l_1 \vee l_2 \vee l_3$
- $(a_1 \vee \neg a_2) \wedge (a_1 \vee b_2 \vee a_2) \wedge a_2$
- $((l_{11} \vee \ldots \vee l_{1m_1}) \wedge \ldots \wedge (l_{n1} \vee \ldots \vee l_{nm_n}))$

Examples DNF

- $\top$
- $\bot$
- a
- $\neg a$
- $l_1 \wedge l_2 \wedge l_3$
- $l_1 \vee l_2 \vee l_3$
- $(a_1 \wedge \neg a_2) \vee (a_1 \wedge b_2 \wedge a_2) \vee a_2$
- $((l_{11} \wedge \ldots \wedge l_{1m_1}) \vee \ldots \vee (l_{n1} \wedge \ldots \wedge l_{nm_n}))$

Representing Functions as CNFs

- **Problem**: Given the truth table of a Boolean function ϕ. How is the function represented in propositional logic?

Solution (in CNF):

1. Represent each assignment ν where ϕ has value **0** as clause:
   - If variable x is **1** in ν, add $\neg x$ to clause.
   - If variable x is **0** in ν, add x to clause.
2. Connect all clauses by conjunction.

a	b	c	ϕ	clauses
0	**0**	**0**	**0**	$a \lor b \lor c$
0	**0**	**1**	**1**	
0	**1**	**0**	**1**	
0	**1**	**1**	**0**	$a \lor \neg b \lor \neg c$
1	**0**	**0**	**1**	
1	**0**	**1**	**0**	$\neg a \lor b \lor \neg c$
1	**1**	**0**	**0**	$\neg a \lor \neg b \lor c$
1	**1**	**1**	**1**	

$\phi =$
$(a \lor b \lor c) \land (a \lor \neg b \lor \neg c) \land (\neg a \lor b \lor \neg c) \land (\neg a \lor \neg b \lor c)$

Representing Functions as DNFs

- **Problem**: Given the truth table of a Boolean function ϕ. How is the function represented in propositional logic?

Solution (in DNF):

1. Represent each assignment ν where ϕ has value **1** as cube:
   - If variable x is **1** in ν, add x to cube.
   - If variable x is **0** in ν, add $\neg x$ to cube.
2. Connect all cubes by disjunction.

a	b	c	ϕ	cubes
0	**0**	**0**	**0**	
0	**0**	**1**	**1**	$\neg a \wedge \neg b \wedge c$
0	**1**	**0**	**1**	$\neg a \wedge b \wedge \neg c$
0	**1**	**1**	**0**	
1	**0**	**0**	**1**	$a \wedge \neg b \wedge \neg c$
1	**0**	**1**	**0**	
1	**1**	**0**	**0**	
1	**1**	**1**	**1**	$a \wedge b \wedge c$

$\phi =$
$(\neg a \wedge \neg b \wedge c) \vee (\neg a \wedge b \wedge \neg c) \vee (a \wedge \neg b \wedge \neg c) \vee$
$(a \wedge b \wedge c)$

PROPOSITIONAL FORMULAS: NORMAL FORM TRANSFORMATION

VL Logic

Part I: Propositional Logic

JOHANNES KEPLER
UNIVERSITY LINZ

Transformation to Conjunctive Normal Form 1

Approach 1: Transformation by “multiplication”

1. Remove $\leftrightarrow, \rightarrow, \oplus$ as follows:
   - $\phi \leftrightarrow \psi \Leftrightarrow (\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$
   - $\phi \rightarrow \psi \Leftrightarrow \neg\phi \vee \psi$,
   - $\phi \oplus \psi \Leftrightarrow (\phi \vee \psi) \wedge (\neg\phi \vee \neg\psi)$
2. Transform formula to negation normal form (NNF) by
   - application of laws of De Morgan
   - elimination of double negation
3. Transform formula to CNF by laws of distributivity

Example: Transformation to CNF 1

Transformation of $\neg(a \leftrightarrow b) \rightarrow (\neg(c \wedge d) \wedge e)$ to an equivalent formula in CNF by approach 1

1. a) remove equivalences: $\Leftrightarrow \neg((a \rightarrow b) \wedge (b \rightarrow a)) \rightarrow (\neg(c \wedge d) \wedge e)$
   b) remove implications: $\Leftrightarrow \neg\neg((\neg a \vee b) \wedge (\neg b \vee a)) \vee (\neg(c \wedge d) \wedge e)$
2. NNF: $\Leftrightarrow ((\neg a \vee b) \wedge (\neg b \vee a)) \vee ((\neg c \vee \neg d) \wedge e)$
3. $\Leftrightarrow ((\neg a \vee b) \vee ((\neg c \vee \neg d) \wedge e)) \wedge ((\neg b \vee a) \vee ((\neg c \vee \neg d) \wedge e)))$
   $\Leftrightarrow (\neg a \vee b \vee \neg c \vee \neg d) \wedge (\neg a \vee b \vee e) \wedge (\neg b \vee a \vee \neg c \vee \neg d) \wedge (\neg b \vee a \vee e)$

Transformation to Conjunctive Normal Form 2

Approach 2: Transformation by introducing labels for subformulas

Given formula ϕ. The following approach transforms ϕ to an equi-satisfiable formula in CNF.

1. Introduce new label ℓ_ψ for each subformula ψ that is not a literal
2. Collect all definitions $\ell_\psi \leftrightarrow \psi'$ in a big conjunction Φ'
   (ψ' is obtained from ψ by replacing its immediate subformulas by the respective labels)
3. Transform Φ' to CNF Φ by approach 1 (no exponential blowup!)

$(\Phi \wedge \ell_\phi)$ **and** ϕ **are equi-satisfiable**

Example: Transformation to CNF 2

Transform $\phi := \neg(a \leftrightarrow b) \rightarrow (\neg(c \wedge d) \wedge e)$ to an equi-satisfiable formula in CNF

definitions Φ'	clauses Φ
$v_1 \leftrightarrow (a \leftrightarrow b)$	$(\bar{v}_1 \vee \bar{a} \vee b), (\bar{v}_1 \vee a \vee \bar{b}), (v_1 \vee \bar{a} \vee \bar{b}), (v_1 \vee a \vee b)$
$v_2 \leftrightarrow \neg v_1$	$(v_1 \vee v_2), (\bar{v}_1 \vee \bar{v}_2)$
$v_3 \leftrightarrow (c \wedge d)$	$(\bar{v}_3 \vee c), (\bar{v}_3 \vee d), (v_3 \vee \bar{c} \vee \bar{d})$
$v_4 \leftrightarrow \neg v_3$	$(v_3 \vee v_4), (\bar{v}_3 \vee \bar{v}_4)$
$v_5 \leftrightarrow (v_4 \wedge e)$	$(\bar{v}_5 \vee v_4), (\bar{v}_5 \vee e), (v_5 \vee \bar{v}_4 \vee \bar{e})$
$v_6 \leftrightarrow (v_2 \rightarrow v_5)$	$(\bar{v}_6 \vee \bar{v}_2 \vee v_5), (v_2 \vee v_6), (\bar{v}_5 \vee v_6)$

$(\Phi \wedge v_6)$ and ϕ are equi-satisfiable.

Some Remarks on Normal Forms

- Approach 1 is exponential in the worst case (e.g., transform $(a_1 \wedge b_1) \vee (a_2 \wedge b_2) \vee \cdots \vee (a_n \wedge b_n)$ to CNF).
- Approach 2 is polynomial
  - Basic idea: introduce labels for subformulas.
  - Also works for formulas with sharing (circuits).
  - Also known as "Tseitin Encoding".
- CNF is usually not easier to solve, but easier to handle:
  - compact data structures: a CNF is simply a list of lists of literals.
- CNF very popular in practice: standard input format DIMACS
- To solve satisfiability of CNF, there are many optimization techniques and dedicated algorithms.

PROPOSITIONAL FORMULAS: RESOLUTION

VL Logic

Part I: Propositional Logic

JOHANNES KEPLER
UNIVERSITY LINZ

Resolution

- the **resolution calculus** consists of the single resolution rule

$$\frac{x \vee C \qquad \neg x \vee D}{C \vee D}$$

  - C and D are (possibly empty) clauses
  - the clause $C \vee D$ is called **resolvent**
  - variable x is called **pivot**
  - usually antecedent clauses $x \vee C$ and $\neg x \vee D$ are assumed not to be tautological
- resolution is **sound** and **complete.**

Example

one application of resolution

$$\frac{x \vee y \vee \neg z \qquad \neg x \vee y' \vee \neg z}{y \vee \neg z \vee y'}$$

derivation of empty clause:

$$\frac{y \qquad \neg y}{\bot}$$

derivation of tautology:

$$\frac{x \vee a \qquad \neg x \vee \neg a}{a \vee \neg a}$$

- the resolution calculus works only on formulas in CNF
- if the empty clause can be derived then the formula is **unsatisfiable**
- if no new clause can be generated by application of the resolution rule then the formula is **satisfiable**

Resolution Example

We **prove unsatisfiability** of

$$\{(\neg x_1 \lor \neg x_5), (x_4 \lor x_5), (x_2 \lor \neg x_4), (x_3 \lor \neg x_4), (\neg x_2 \lor \neg x_3), (x_1 \lor x_4 \lor \neg x_6), (x_6)\}$$

as follows: