Theory of Computer Science

B2. Propositional Logic II

Malte Helmert

University of Basel

March 1, 2017

Theory of Computer Science
March 1, 2017 — B2. Propositional Logic II

B2.1 Equivalences

Equivalent Formulas

Definition (Equivalence of Propositional Formulas)

Two propositional formulas φ and ψ over A are (logically) equivalent ($\varphi \equiv \psi$) if for all interpretations $\mathcal{I}$ for A it is true that $\mathcal{I} \models \varphi$ if and only if $\mathcal{I} \models \psi$.

German: logisch äquivalent

Equivalent Formulas: Example

$$((\varphi \lor \psi) \lor \chi) \equiv (\varphi \lor (\psi \lor \chi))$$

$\mathcal{I} \models \varphi$	$\mathcal{I} \models \psi$	$\mathcal{I} \models \chi$	$\mathcal{I} \models (\varphi \lor \psi)$	$\mathcal{I} \models (\psi \lor \chi)$	$\mathcal{I} \models ((\varphi \lor \psi) \lor \chi)$	$\mathcal{I} \models (\varphi \lor (\psi \lor \chi))$
No	No	No	No	No	No	No
No	No	Yes	No	Yes	Yes	Yes
No	Yes	No	Yes	Yes	Yes	Yes
No	Yes	Yes	Yes	Yes	Yes	Yes
Yes	No	No	Yes	No	Yes	Yes
Yes	No	Yes	Yes	Yes	Yes	Yes
Yes	Yes	No	Yes	Yes	Yes	Yes
Yes	Yes	Yes	Yes	Yes	Yes	Yes

Some Equivalences (1)

$$(\varphi \land \varphi) \equiv \varphi$$
$$(\varphi \lor \varphi) \equiv \varphi \qquad \text{(idempotence)}$$

$$(\varphi \land \psi) \equiv (\psi \land \varphi)$$
$$(\varphi \lor \psi) \equiv (\psi \lor \varphi) \qquad \text{(commutativity)}$$

$$((\varphi \land \psi) \land \chi) \equiv (\varphi \land (\psi \land \chi))$$
$$((\varphi \lor \psi) \lor \chi) \equiv (\varphi \lor (\psi \lor \chi)) \qquad \text{(associativity)}$$

German: Idempotenz, Kommutativität, Assoziativität

Some Equivalences (2)

$$(\varphi \land (\varphi \lor \psi)) \equiv \varphi$$
$$(\varphi \lor (\varphi \land \psi)) \equiv \varphi \qquad \text{(absorption)}$$

$$(\varphi \land (\psi \lor \chi)) \equiv ((\varphi \land \psi) \lor (\varphi \land \chi))$$
$$(\varphi \lor (\psi \land \chi)) \equiv ((\varphi \lor \psi) \land (\varphi \lor \chi)) \qquad \text{(distributivity)}$$

German: Absorption, Distributivität

Some Equivalences (3)

$$\lnot\lnot\varphi \equiv \varphi \qquad \text{(Double negation)}$$

$$\lnot(\varphi \land \psi) \equiv (\lnot\varphi \lor \lnot\psi)$$
$$\lnot(\varphi \lor \psi) \equiv (\lnot\varphi \land \lnot\psi) \qquad \text{(De Morgan's rules)}$$

$$(\varphi \lor \psi) \equiv \varphi \text{ if } \varphi \text{ tautology}$$
$$(\varphi \land \psi) \equiv \psi \text{ if } \varphi \text{ tautology} \qquad \text{(tautology rules)}$$

$$(\varphi \lor \psi) \equiv \psi \text{ if } \varphi \text{ unsatisfiable}$$
$$(\varphi \land \psi) \equiv \varphi \text{ if } \varphi \text{ unsatisfiable} \qquad \text{(unsatisfiability rules)}$$

German: Doppelnegation, De Morgansche Regeln, Tautologieregeln, Unerfüllbarkeitsregeln

Substitution Theorem

Theorem (Substitution Theorem)

Let φ and φ' be equivalent propositional formulas over A.
Let ψ be a propositional formula with (at least) one occurrence of the subformula φ.

Then ψ is equivalent to ψ', where ψ' is constructed from ψ by replacing an occurrence of φ in ψ with φ'.

German: Ersetzbarkeitstheorem

(without proof)

Application of Equivalences: Example

$$\begin{aligned}(P \land (\neg Q \lor P)) &\equiv ((P \land \neg Q) \lor (P \land P)) && \text{(distributivity)}\\ &\equiv ((P \land \neg Q) \lor P) && \text{(idempotence)}\\ &\equiv (P \lor (P \land \neg Q)) && \text{(commutativity)}\\ &\equiv P && \text{(absorption)}\end{aligned}$$

B2.2 Simplified Notation

Parentheses

Associativity:

$$((\varphi \land \psi) \land \chi) \equiv (\varphi \land (\psi \land \chi))$$
$$((\varphi \lor \psi) \lor \chi) \equiv (\varphi \lor (\psi \lor \chi))$$

- Placement of parentheses for a conjunction of conjunctions does not influence whether an interpretation is a model.
- ditto for disjunctions of disjunctions

$\rightsquigarrow$ can omit parentheses and treat this as if parentheses placed arbitrarily

- Example: $(A_1 \land A_2 \land A_3 \land A_4)$ instead of $((A_1 \land (A_2 \land A_3)) \land A_4)$
- Example: $(\neg A \lor (B \land C) \lor D)$ instead of $((\neg A \lor (B \land C)) \lor D)$

Parentheses

Does this mean we can always omit all parentheses and assume an arbitrary placement? → No!

$$((\varphi \land \psi) \lor \chi) \not\equiv (\varphi \land (\psi \lor \chi))$$

What should $\varphi \land \psi \lor \chi$ mean?

Placement of Parentheses by Convention

Often parentheses can be dropped in specific cases and an implicit placement is assumed:

- $\neg$ binds more strongly than $\land$
- $\land$ binds more strongly than $\lor$
- $\lor$ binds more strongly than $\rightarrow$ or $\leftrightarrow$

⇝ cf. PEMDAS/"Punkt vor Strich"

Example

$A \lor \neg C \land B \rightarrow A \lor \neg D$ stands for $((A \lor (\neg C \land B)) \rightarrow (A \lor \neg D))$

- often harder to read
- error-prone

⇝ not used in this course

Short Notations for Conjunctions and Disjunctions

short notation for addition:

$$\sum_{i=1}^{n} x_i = x_1 + x_2 + \cdots + x_n$$
$$\sum_{x \in \{x_1,\dots,x_n\}} x = x_1 + x_2 + \cdots + x_n$$

Analogously:

$$(\bigwedge_{i=1}^{n} \varphi_i) = (\varphi_1 \land \varphi_2 \land \cdots \land \varphi_n)$$
$$(\bigvee_{i=1}^{n} \varphi_i) = (\varphi_1 \lor \varphi_2 \lor \cdots \lor \varphi_n)$$
$$(\bigwedge_{\varphi \in X} \varphi) = (\varphi_1 \land \varphi_2 \land \cdots \land \varphi_n)$$
$$(\bigvee_{\varphi \in X} \varphi) = (\varphi_1 \lor \varphi_2 \lor \cdots \lor \varphi_n)$$
$$\text{for } X = \{\varphi_1, \dots, \varphi_n\}$$

Short Notation: Corner Cases

Is $\mathcal{I} \models \psi$ true for

$$\psi = (\bigwedge_{\varphi \in X} \varphi) \text{ and } \psi = (\bigvee_{\varphi \in X} \varphi)$$

if $X = \emptyset$ or $X = \{\chi\}$?

convention:

- $(\bigwedge_{\varphi \in \emptyset} \varphi)$ is tautology.
- $(\bigvee_{\varphi \in \emptyset} \varphi)$ is unsatisfiable.
- $(\bigwedge_{\varphi \in \{\chi\}} \varphi) = (\bigvee_{\varphi \in \{\chi\}} \varphi) = \chi$

⇝ Why?

B2.3 Normal Forms

Why Normal Forms?

- A normal form is a representation with certain syntactic restrictions.
- condition for reasonable normal form: every formula must have a logically equivalent formula in normal form
- advantages:
  - can restrict proofs to formulas in normal form
  - can define algorithms only for formulas in normal form

German: Normalform

Literals, Clauses and Monomials

- A literal is an atomic proposition or the negation of an atomic proposition (e. g., A and $\neg A$).
- A clause is a disjunction of literals (e. g., $(Q \lor \neg P \lor \neg S \lor R)$).
- A monomial is a conjunction of literals (e. g., $(Q \land \neg P \land \neg S \land R)$).

The terms clause and monomial are also used for the corner case with only one literal.

German: Literal, Klausel, Monom

Terminology: Examples

Examples

- $(\neg Q \land R)$ is a monomial
- $(P \lor \neg Q)$ is a clause
- $((P \lor \neg Q) \land P)$ is neither literal nor clause nor monomial
- $\neg P$ is a literal, a clause and a monomial
- $(P \rightarrow Q)$ is neither literal nor clause nor monomial (but $(\neg P \lor Q)$ is a clause!)
- $(P \lor P)$ is a clause, but not a literal or monomial
- $\neg\neg P$ is neither literal nor clause nor monomial

Conjunctive Normal Form

Definition (Conjunctive Normal Form)

A formula is in conjunctive normal form (CNF)
if it is a conjunction of clauses, i. e., if it has the form

$$\left(\bigwedge_{i=1}^{n}\left(\bigvee_{j=1}^{m_i} L_{ij}\right)\right)$$

with $n, m_i > 0$ (for $1 \leq i \leq n$), where the L_{ij} are literals.

German: konjunktive Normalform (KNF)

Example

((¬P ∨ Q) ∧ R ∧ (P ∨ ¬S)) is in CNF.

Disjunctive Normal Form

Definition (Disjunctive Normal Form)

A formula is in disjunctive normal form (DNF)
if it is a disjunction of monomials, i. e., if it has the form

$$\left(\bigvee_{i=1}^{n}\left(\bigwedge_{j=1}^{m_i} L_{ij}\right)\right)$$

with $n, m_i > 0$ (for $1 \leq i \leq n$), where the L_{ij} are literals.

German: disjunktive Normalform (DNF)

Example

((¬P ∧ Q) ∨ R ∨ (P ∧ ¬S)) is in DNF.

CNF and DNF: Examples

Examples

- ((P ∨ ¬Q) ∧ P) is in CNF
- ((R ∨ Q) ∧ P ∧ (R ∨ S)) is in CNF
- (P ∨ (¬Q ∧ R)) is in DNF
- ((P ∨ ¬Q) → P) is neither in CNF nor in DNF
- P is in CNF and in DNF

Construction of CNF (and DNF)

Algorithm to Construct CNF

1. Replace abbreviations → and ↔ by their definitions
((→)-elimination and (↔)-elimination).
⇝ formula structure: only ∨, ∧, ¬
2. Move negations inside using De Morgan and double negation.
⇝ formula structure: only ∨, ∧, literals
3. Distribute ∨ over ∧ with distributivity
(strictly speaking also with commutativity).
⇝ formula structure: CNF
4. optionally: Simplify the formula at the end
or at intermediate steps (e. g., with idempotence).

Note: For DNF, distribute ∧ over ∨ instead.
Question: runtime complexity?

Constructing CNF: Example

Construction of Conjunctive Normal Form

Given: $\varphi = (((P \land \neg Q) \lor R) \rightarrow (P \lor \neg(S \lor T)))$

$$\begin{aligned}
\varphi &\equiv (\neg((P \land \neg Q) \lor R) \lor P \lor \neg(S \lor T)) && \text{[Step 1]}\\
&\equiv ((\neg(P \land \neg Q) \land \neg R) \lor P \lor \neg(S \lor T)) && \text{[Step 2]}\\
&\equiv (((\neg P \lor \neg\neg Q) \land \neg R) \lor P \lor \neg(S \lor T)) && \text{[Step 2]}\\
&\equiv (((\neg P \lor Q) \land \neg R) \lor P \lor \neg(S \lor T)) && \text{[Step 2]}\\
&\equiv (((\neg P \lor Q) \land \neg R) \lor P \lor (\neg S \land \neg T)) && \text{[Step 2]}\\
&\equiv ((\neg P \lor Q \lor P \lor (\neg S \land \neg T)) \land\\
&\quad\ (\neg R \lor P \lor (\neg S \land \neg T))) && \text{[Step 3]}\\
&\equiv (\neg R \lor P \lor (\neg S \land \neg T)) && \text{[Step 4]}\\
&\equiv ((\neg R \lor P \lor \neg S) \land (\neg R \lor P \lor \neg T)) && \text{[Step 3]}
\end{aligned}$$

Construct DNF: Example

Construction of Disjunctive Normal Form

Given: $\varphi = (((P \land \neg Q) \lor R) \rightarrow (P \lor \neg(S \lor T)))$

$$\begin{aligned}
\varphi &\equiv (\neg((P \land \neg Q) \lor R) \lor P \lor \neg(S \lor T)) && \text{[Step 1]}\\
&\equiv ((\neg(P \land \neg Q) \land \neg R) \lor P \lor \neg(S \lor T)) && \text{[Step 2]}\\
&\equiv (((\neg P \lor \neg\neg Q) \land \neg R) \lor P \lor \neg(S \lor T)) && \text{[Step 2]}\\
&\equiv (((\neg P \lor Q) \land \neg R) \lor P \lor \neg(S \lor T)) && \text{[Step 2]}\\
&\equiv (((\neg P \lor Q) \land \neg R) \lor P \lor (\neg S \land \neg T)) && \text{[Step 2]}\\
&\equiv ((\neg P \land \neg R) \lor (Q \land \neg R) \lor P \lor (\neg S \land \neg T)) && \text{[Step 3]}
\end{aligned}$$

Existence of an Equivalent Formula in Normal Form

Theorem

For every formula φ there is a logically equivalent formula in CNF and a logically equivalent formula in DNF.

- "There is a" always means "there is at least one". Otherwise we would write "there is exactly one".
- Intuition: algorithm to construct normal form works with any given formula and only uses equivalence rewriting.
- actual proof would use induction over structure of formula

More Theorems

Theorem

A formula in CNF is a tautology iff every clause is a tautology.

Theorem

A formula in DNF is satisfiable iff at least one its monomials is satisfiable.

$\rightsquigarrow$ both proved easily with semantics of propositional logic

B2.4 Logical Consequences

Knowledge Bases: Example

If not DrinkBeer, then EatFish.
If EatFish and DrinkBeer,
then not EatIceCream.
If EatIceCream or not DrinkBeer,
then not EatFish.

$$\begin{aligned}\text{KB} = \{&(\neg\text{DrinkBeer} \rightarrow \text{EatFish}),\\ &((\text{EatFish} \land \text{DrinkBeer}) \rightarrow \neg\text{EatIceCream}),\\ &((\text{EatIceCream} \lor \neg\text{DrinkBeer}) \rightarrow \neg\text{EatFish})\}\end{aligned}$$

Exercise from U. Schöning: Logik für Informatiker
Picture courtesy of graur razvan ionut / FreeDigitalPhotos.net

Models for Sets of Formulas

Definition (Model for Knowledge Base)

Let KB be a knowledge base over A,
i. e., a set of propositional formulas over A.

A truth assignment $\mathcal{I}$ for A is a model for KB (written: $\mathcal{I} \models \text{KB}$)
if $\mathcal{I}$ is a model for every formula $\varphi \in \text{KB}$.

German: Wissensbasis, Modell

Properties of Sets of Formulas

A knowledge base KB is

- satisfiable if KB has at least one model
- unsatisfiable if KB is not satisfiable
- valid (or a tautology) if every interpretation is a model for KB
- falsifiable if KB is no tautology

German: erfüllbar, unerfüllbar, gültig, gültig/eine Tautologie, falsifizierbar

Example I

Which of the properties does $\mathrm{KB} = \{(A \land \neg B), \neg(B \lor A)\}$ have?

KB is unsatisfiable:
For every model $\mathcal{I}$ with $\mathcal{I} \models (A \land \neg B)$ we have $\mathcal{I}(A) = 1$.
This means $\mathcal{I} \models (B \lor A)$ and thus $\mathcal{I} \not\models \neg(B \lor A)$.

This directly implies that KB is falsifiable, not satisfiable and no tautology.

Example II

Which of the properties does

$$\mathrm{KB} = \{(\neg \mathrm{DrinkBeer} \rightarrow \mathrm{EatFish}), ((\mathrm{EatFish} \land \mathrm{DrinkBeer}) \rightarrow \neg \mathrm{EatIceCream}), ((\mathrm{EatIceCream} \lor \neg \mathrm{DrinkBeer}) \rightarrow \neg \mathrm{EatFish})\}$$

have?

- satisfiable, e. g. with $\mathcal{I} = \{\mathrm{EatFish} \mapsto 1, \mathrm{DrinkBeer} \mapsto 1, \mathrm{EatIceCream} \mapsto 0\}$
- thus not unsatisfiable
- falsifiable, e. g. with $\mathcal{I} = \{\mathrm{EatFish} \mapsto 0, \mathrm{DrinkBeer} \mapsto 0, \mathrm{EatIceCream} \mapsto 1\}$
- thus not valid

Logical Consequences: Motivation

What's the secret of your long life?

I am on a strict diet: If I don't drink beer to a meal, then I always eat fish. Whenever I have fish and beer with the same meal, I abstain from ice cream. When I eat ice cream or don't drink beer, then I never touch fish.

Claim: the woman drinks beer to every meal.

How can we prove this?

Exercise from U. Schöning: Logik für Informatiker
Picture courtesy of graur razvan ionut/FreeDigitalPhotos.net

Logical Consequences

Definition (Logical Consequence)

Let KB be a set of formulas and φ a formula.

We say that KB logically implies φ (written as $\mathrm{KB} \models \varphi$) if all models of KB are also models of φ.

also: KB logically entails φ, φ logically follows from KB, φ is a logical consequence of KB

German: KB impliziert φ logisch, φ folgt logisch aus KB, φ ist logische Konsequenz von KB

Attention: the symbol $\models$ is "overloaded": $\mathrm{KB} \models \varphi$ vs. $\mathcal{I} \models \varphi$.

What if KB is unsatisfiable or the empty set?

Logical Consequences: Example

Let φ = DrinkBeer and

$$\begin{aligned} \text{KB} = \{&(\neg\text{DrinkBeer} \rightarrow \text{EatFish}),\\ &((\text{EatFish} \land \text{DrinkBeer}) \rightarrow \neg\text{EatIceCream}),\\ &((\text{EatIceCream} \lor \neg\text{DrinkBeer}) \rightarrow \neg\text{EatFish})\}. \end{aligned}$$

Show: KB $\models \varphi$

Proof sketch.

Proof by contradiction: assume $\mathcal{I} \models$ KB, but $\mathcal{I} \not\models$ DrinkBeer.
Then it follows that $\mathcal{I} \models \neg$DrinkBeer.
Because $\mathcal{I}$ is a model of KB, we also have
$\mathcal{I} \models (\neg\text{DrinkBeer} \rightarrow \text{EatFish})$ and thus $\mathcal{I} \models$ EatFish. (Why?)
With an analogous argumentation starting from
$\mathcal{I} \models ((\text{EatIceCream} \lor \neg\text{DrinkBeer}) \rightarrow \neg\text{EatFish})$
we get $\mathcal{I} \models \neg$EatFish and thus $\mathcal{I} \not\models$ EatFish. $\rightsquigarrow$ Contradiction!

Important Theorems about Logical Consequences

Theorem (Deduction Theorem)

KB $\cup \{\varphi\} \models \psi$ *iff* KB $\models (\varphi \rightarrow \psi)$

German: Deduktionssatz

Theorem (Contraposition Theorem)

KB $\cup \{\varphi\} \models \neg\psi$ *iff* KB $\cup \{\psi\} \models \neg\varphi$

German: Kontrapositionssatz

Theorem (Contradiction Theorem)

KB $\cup \{\varphi\}$ *is unsatisfiable iff* KB $\models \neg\varphi$

German: Widerlegungssatz

(without proof)

B2.5 Summary

Summary

- Logical equivalence describes when formulas are semantically indistinguishable.
- Equivalence rewriting is used to simplify formulas and to bring them in normal forms.
- CNF: formula is a conjunction of clauses
- DNF: formula is a disjunction of monomials
- every formula has equivalent formulas in DNF and in CNF
- knowledge base: set of formulas describing given information; satisfiable, valid etc. used like for individual formulas
- logical consequence KB $\models \varphi$ means that φ is true whenever (= in all models where) KB is true