Formal Methods in software development

a.y.2017/2018

Prof. Anna Labella

Completeness of natural deduction

$$\varphi_1, \varphi_2, \varphi_3, \ldots.. \vdash \psi$$

$$\uparrow$$

$$\varphi_1, \varphi_2, \varphi_3, \ldots.. \models \psi$$

Completeness of natural deduction: proof 1

First step:

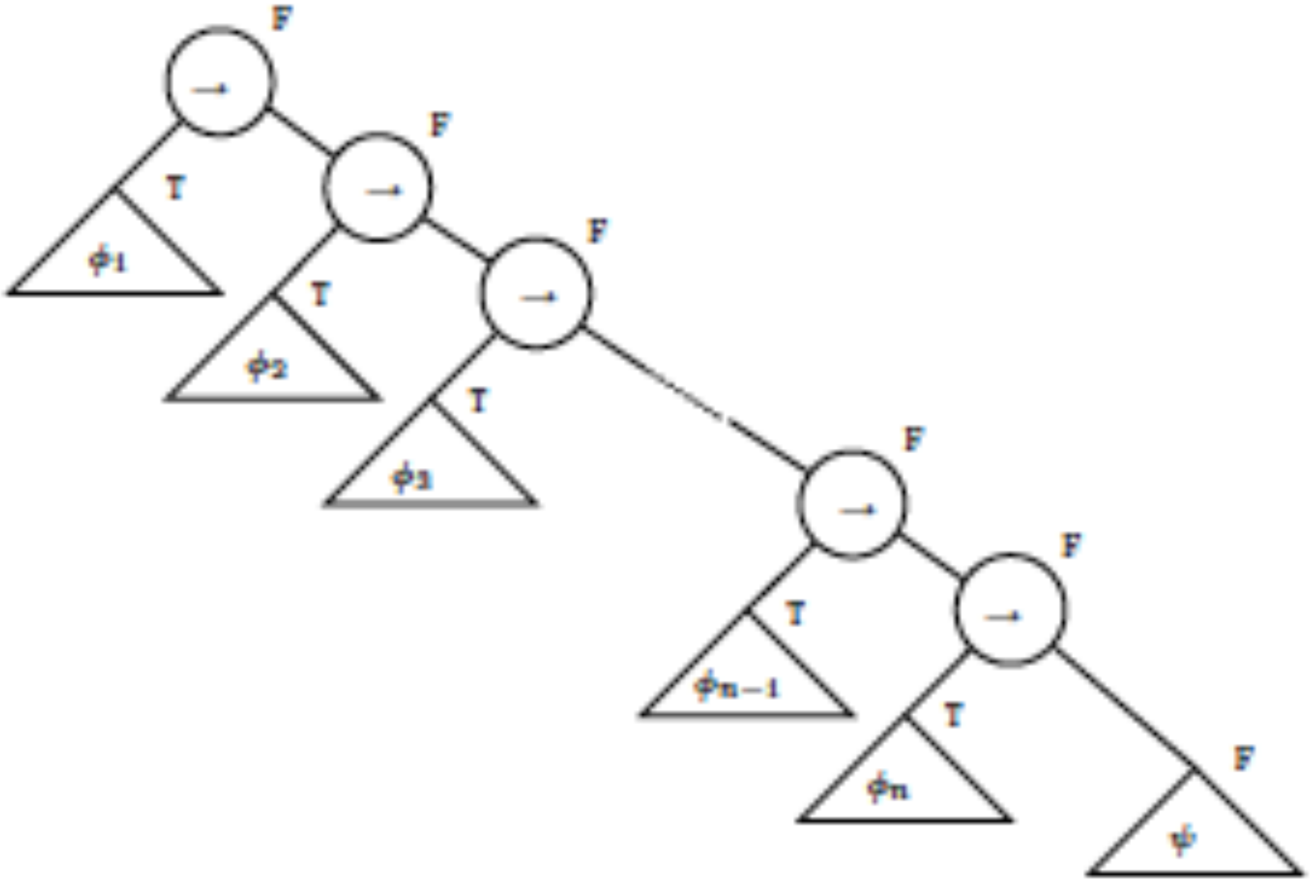
$$\varphi_1, \varphi_2, \varphi_3, \ldots.. \;|\!==\quad \psi$$

iff

$$|\!== (\varphi_1 \rightarrow\; (\varphi_2 \rightarrow (\; \varphi_3 \rightarrow\; \ldots.. \rightarrow\; \psi)))$$

tautology

Completeness of natural deduction: proof



Completeness of natural deduction: proof 2

Second step:

$|\!== (\varphi_1 \to \ (\varphi_2 \to (\ \varphi_3 \to \ \ldots.. \to \ \psi)))$
tautology

↓

$|\text{--} \ (\varphi_1 \to \ (\varphi_2 \to (\ \varphi_3 \to \ \ldots.. \to \ \psi)))$
proof

Completeness of natural deduction: proof 3

Let us code the lines of truth table for a formula φ, using its atoms
($\underline{p}_i$ is p_i or $\neg p_i$ according to its value)

Proposition
$\underline{p}_1, \underline{p}_2$, …$\underline{p}_n$ |-φ for every line producing true
$\underline{p}_1, \underline{p}_2$, …$\underline{p}_n$ |- ¬φ for every line producing false

Proof by structural induction on φ

Completeness of natural deduction: proof 4

1. φ is an atom p then we have $p \vdash p$ and $\neg p \vdash \neg p$
2. φ is $\neg \varphi_1$
3. φ is $\varphi_1 \rightarrow \varphi_2$
4. φ is $\varphi_1 \wedge \varphi_2$
5. φ is $\varphi_1 \vee \varphi_2$

Let us observe that $\underline{p_1}, \underline{p_2}, \ldots \underline{p_k} \vdash \varphi_1$ and $\underline{p_{k+1}}, \underline{p_{k+2}}, \ldots \underline{p_n} \vdash \varphi_2$ implies

$\underline{p_1}, \underline{p_2}, \ldots \underline{p_n} \vdash \varphi_1 \wedge \varphi_2$

Completeness of natural deduction: proof 5

To prove the proposition for

2. φ is $\varphi_1 \rightarrow \varphi_2$ is equivalent to say

$\varphi_1 \land \neg\varphi_2 \vdash \neg(\varphi_1 \rightarrow \varphi_2)$ the only false case

$\neg\varphi_1 \land \neg\varphi_2 \vdash \varphi_1 \rightarrow \varphi_2$

$\neg\varphi_1 \land \varphi_2 \vdash \varphi_1 \rightarrow \varphi_2$

$\varphi_1 \land \varphi_2 \vdash \varphi_1 \rightarrow \varphi_2$

These are all the possible cases

Completeness of natural deduction: proof 6

And so on for the other connectives

3. φ is $\varphi_1 \wedge \varphi_2$

only one true case

$\varphi_1 \wedge \neg\varphi_2 \vdash \neg(\varphi_1 \wedge \varphi_2)$

$\neg\varphi_1 \wedge \varphi_2 \vdash \neg(\varphi_1 \wedge \varphi_2)$

$\neg\varphi_1 \wedge \neg \varphi_2 \vdash \neg(\varphi_1 \wedge \varphi_2)$

4. φ is $\varphi_1 \vee \varphi_2$

$\neg \varphi_1 \wedge \neg\varphi_2 \vdash \neg(\varphi_1 \vee \varphi_2)$ the only false case

$\varphi_1 \wedge \neg\varphi_2 \vdash \varphi_1 \vee \varphi_2$

$\neg\varphi_1 \wedge \varphi_2 \vdash \varphi_1 \vee \varphi_2$

$\varphi_1 \wedge \varphi_2 \vdash \varphi_1 \vee \varphi_2$

Completeness of natural deduction: proof 7

If we apply the above proposition to

$$|\!== (\varphi_1 \to (\varphi_2 \to (\varphi_3 \to \ldots.. \to \psi)))$$

We have 2^n proofs

$$\underline{p}_1, \underline{p}_2, \ldots \underline{p}_n \vdash (\varphi_1 \to (\varphi_2 \to (\varphi_3 \to \ldots.. \to \psi)))$$

Let us eliminate all the premises, because they are pairwise complementary, by using tertium non datur LEM as in the scheme

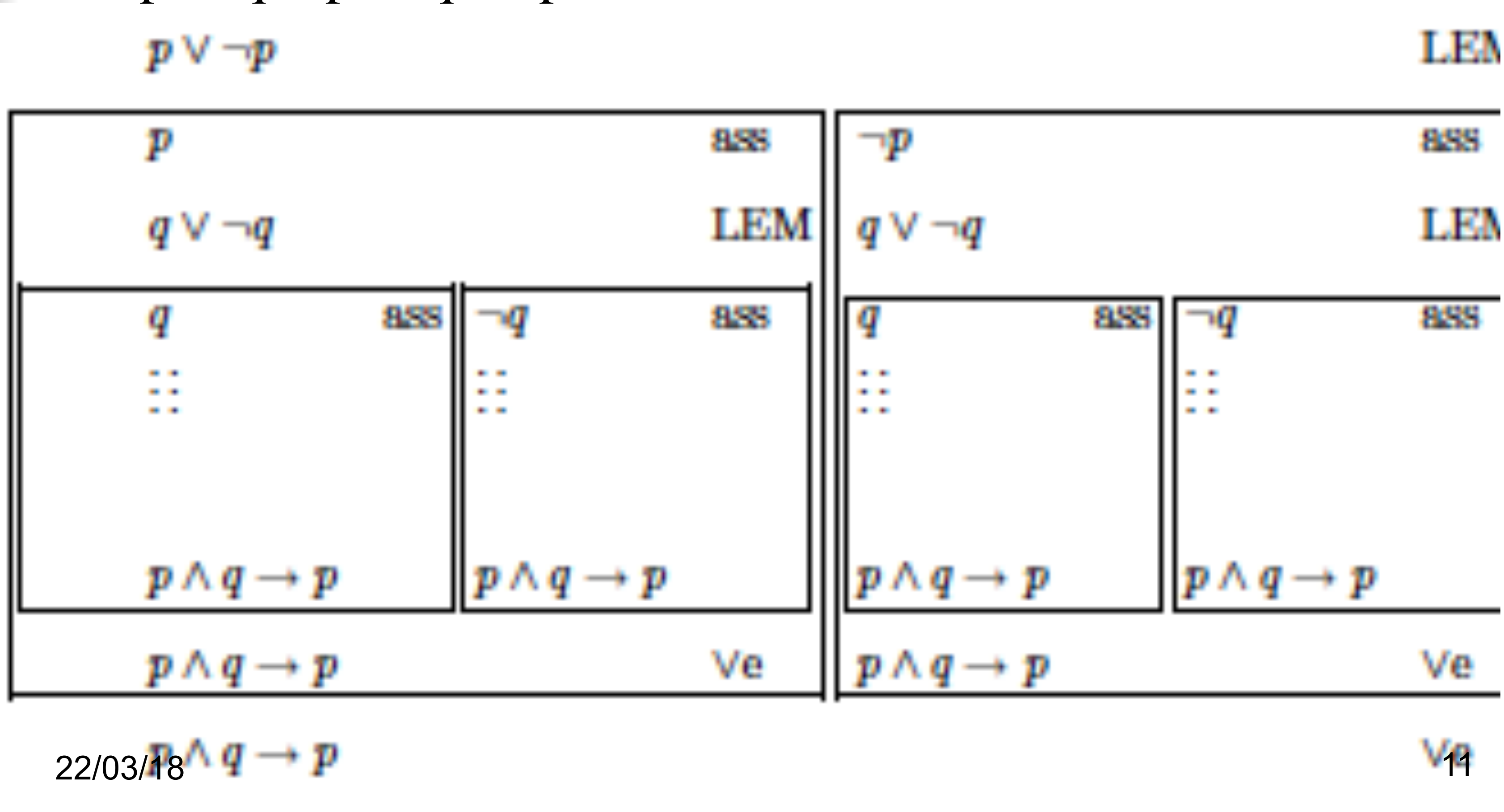
Example: let us take the tautology $p \wedge q \to p$, we have

$p, q \vdash p \wedge q \to p$

$p, \neg q \vdash p \wedge q \to p$

$\neg p, q \vdash p \wedge q \to p$

$\neg p, \neg q \vdash p \wedge q \to p$



$p \vee \neg p$ LEM

p ass

$q \vee \neg q$ LEM

q ass

$\vdots$

$p \wedge q \to p$

$\neg q$ ass

$\vdots$

$p \wedge q \to p$

$p \wedge q \to p$ ∨e

$\neg p$ ass

$q \vee \neg q$ LEM

q ass

$\vdots$

$p \wedge q \to p$

$\neg q$ ass

$\vdots$

$p \wedge q \to p$

$p \wedge q \to p$ ∨e

$p \wedge q \to p$ ∨e

Completeness of natural deduction: proof 8

Third step:

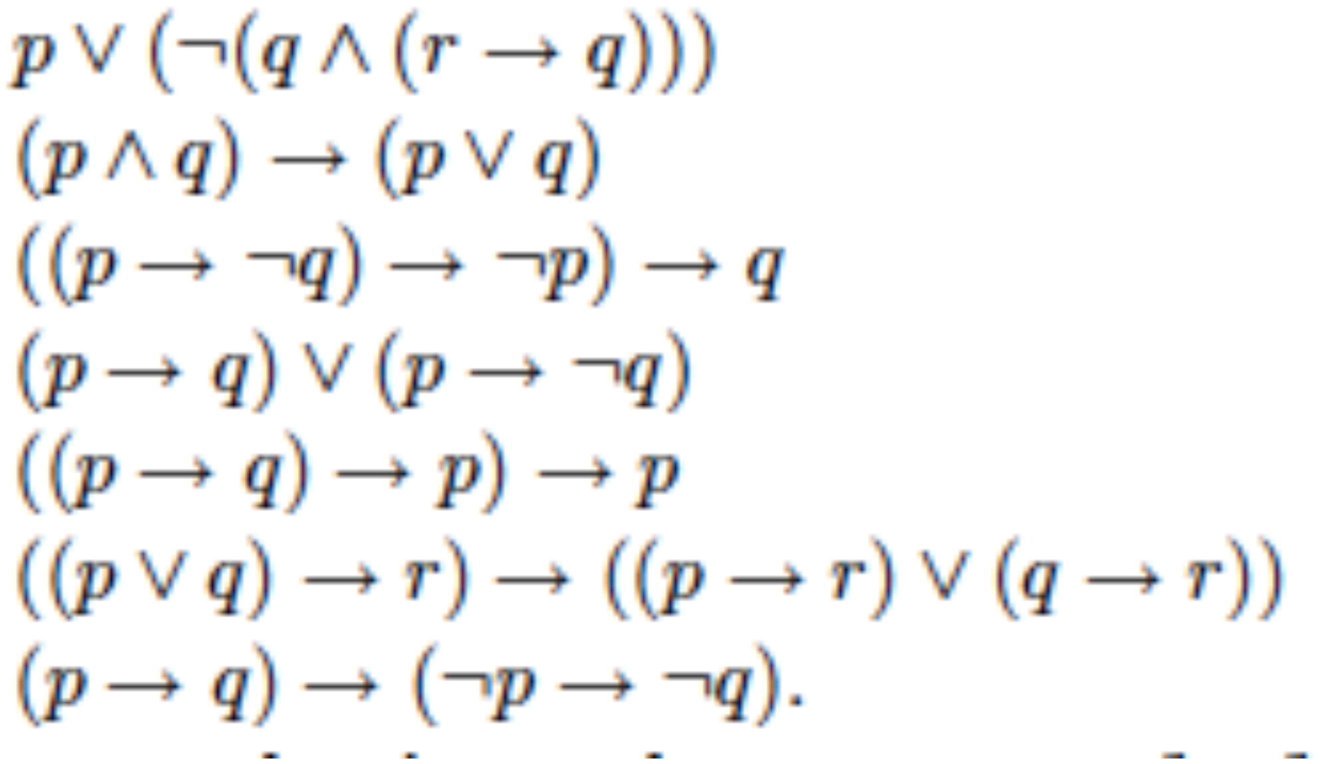
$\vdash (\varphi_1 \rightarrow (\varphi_2 \rightarrow (\varphi_3 \rightarrow \ldots\ldots \rightarrow \psi)))$

$\Downarrow$

$\varphi_1, \varphi_2, \varphi_3, \ldots\ldots \vdash \psi$

But we know that this is true

Truth tables exercises



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

$$(p \wedge q) \rightarrow (p \vee q)$$

$$((p \rightarrow \neg q) \rightarrow \neg p) \rightarrow q$$

$$(p \rightarrow q) \vee (p \rightarrow \neg q)$$

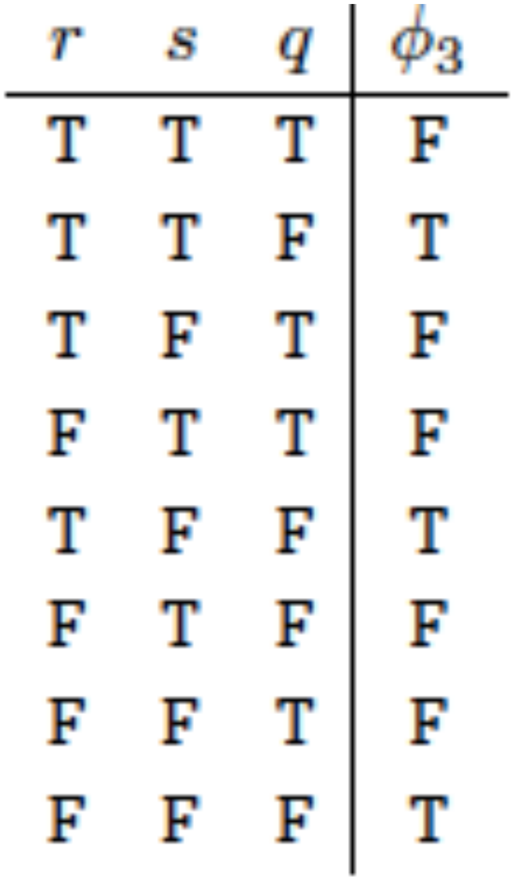
$$((p \rightarrow q) \rightarrow p) \rightarrow p$$

$$((p \vee q) \rightarrow r) \rightarrow ((p \rightarrow r) \vee (q \rightarrow r))$$

$$(p \rightarrow q) \rightarrow (\neg p \rightarrow \neg q).$$

Truth tables exercises

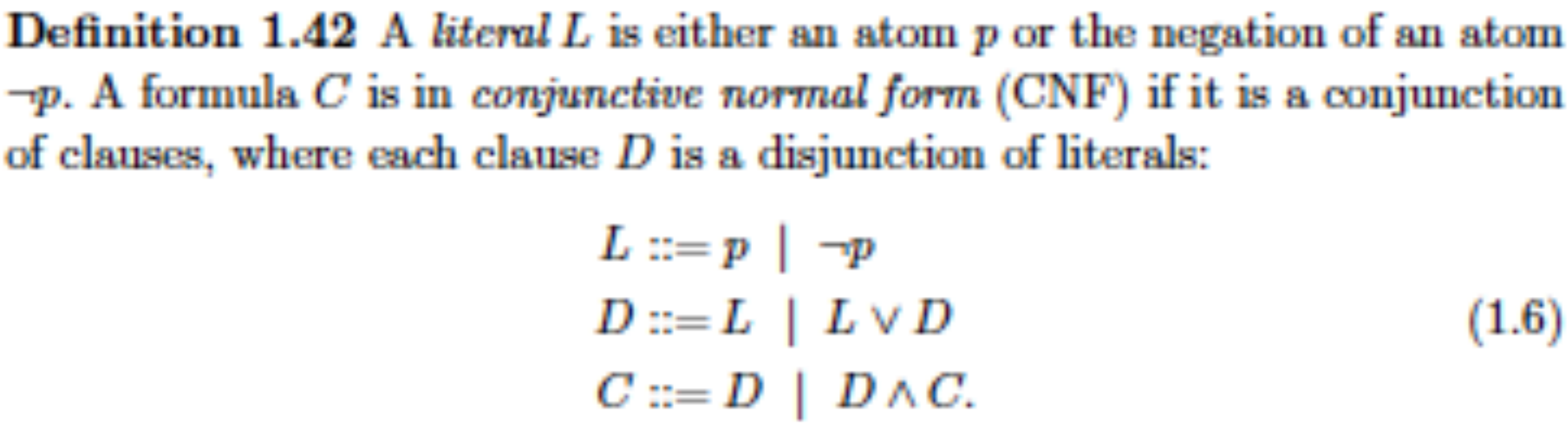
Who φ_3 is?



r	s	q	ϕ_3
T	T	T	F
T	T	F	T
T	F	T	F
F	T	T	F
T	F	F	T
F	T	F	F
F	F	T	F
F	F	F	T

Validity and satisfiability: CNF

A formula is valid when it is true for any assignment
It is satisfiable when it is true for at least one assignment



Definition 1.42 A *literal* L is either an atom p or the negation of an atom $\neg p$. A formula C is in *conjunctive normal form* (CNF) if it is a conjunction of clauses, where each clause D is a disjunction of literals:

$$\begin{aligned} L &::= p \mid \neg p \\ D &::= L \mid L \vee D \\ C &::= D \mid D \wedge C. \end{aligned} \tag{1.6}$$

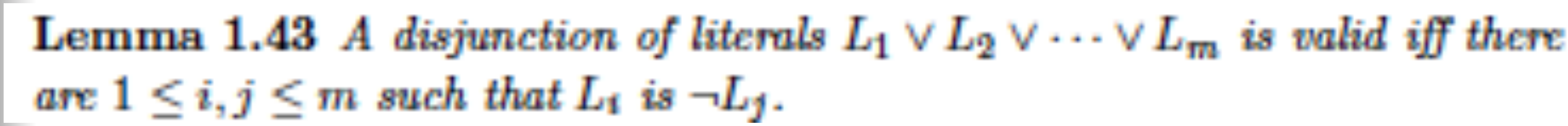
Validity and satisfiability: CNF

In a CNF looking for validity
means to check validity of every conjunct;

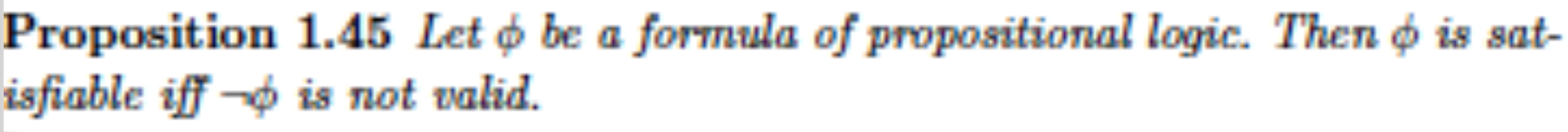
this means that every conjunct must contain
a pair of opposite literals

This is often an efficient way of checking validity

CNF



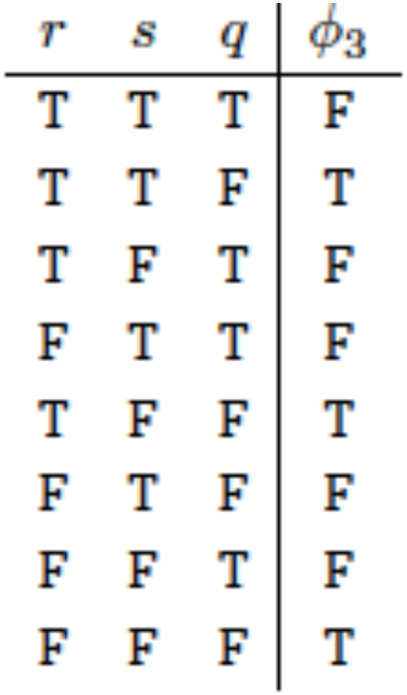
Lemma 1.43 *A disjunction of literals* $L_1 \vee L_2 \vee \cdots \vee L_m$ *is valid iff there are* $1 \leq i, j \leq m$ *such that* L_i *is* $\neg L_j$.



Proposition 1.45 *Let* ϕ *be a formula of propositional logic. Then* ϕ *is satisfiable iff* $\neg\phi$ *is not valid.*

CNF

In the case we are given with a truth table, we can compute the CNF directly



r	s	q	ϕ_3
T	T	T	F
T	T	F	T
T	F	T	F
F	T	T	F
T	F	F	T
F	T	F	F
F	F	T	F
F	F	F	T

CNF

Let us take all the false cases, namely lines 1, 3, 4, 6 and 7

None of them can happen if the formula has to be true

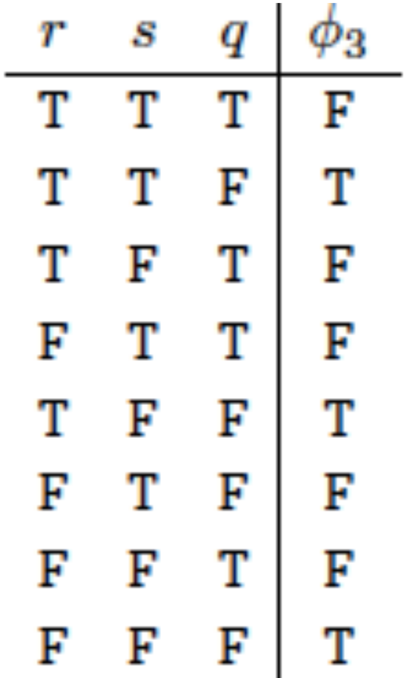
$\neg((r \wedge s \wedge q) \vee (r \wedge \neg s \wedge q) \vee (\neg r \wedge s \wedge q) \vee (\neg r \wedge s \wedge \neg q) \vee (\neg r \wedge \neg s \wedge q)\ )$

i.e. via De Morgan

$\neg(r \wedge s \wedge q) \wedge \neg(r \wedge \neg s \wedge q) \wedge \neg(\neg r \wedge s \wedge q) \wedge \neg(\neg r \wedge s \wedge \neg q) \wedge \neg(\neg r \wedge \neg s \wedge q)$

i.e. via De Morgan

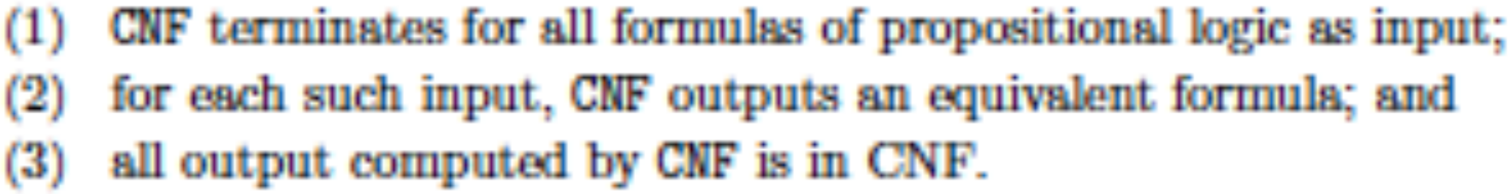
$(\neg r \vee \neg s \vee \neg q) \wedge (\neg r \vee s \vee \neg q) \wedge (r \vee \neg s \vee \neg q) \wedge (r \vee \neg s \vee q) \wedge (r \vee s \vee \neg q)$



r	s	q	ϕ_3
T	T	T	F
T	T	F	T
T	F	T	F
F	T	T	F
T	F	F	T
F	T	F	F
F	F	T	F
F	F	F	T

CNF

If we are given with the synctactical expression, we define an algorithm CNF such that



(1) CNF terminates for all formulas of propositional logic as input;
(2) for each such input, CNF outputs an equivalent formula; and
(3) all output computed by CNF is in CNF.

Neither efficiency nor unicity is secured

e.g. p and $p \wedge (p \vee q)$

CNF: the algorithm

- IMPL_FREE eliminate implications
- NNF pull inside negations
- DISTR Uses distributivity

CNF: the algorithm

- IMPL_FREE to eliminate implications

$p \rightarrow q$ is equivalent to $\neg p \vee q$

- NNF to pull inside negations

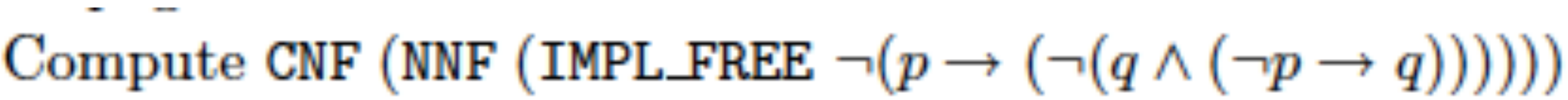
$\neg(p \vee q)$ is equivalent to $\neg p \wedge \neg q$

$\neg(p \wedge q)$ is equivalent to $\neg p \vee \neg q$

- DISTR uses distributivity to extract $\wedge$

$r \vee (p \wedge q)$ is equivalent to $(r \vee p) \wedge (r \vee q)$

CNF: exercises



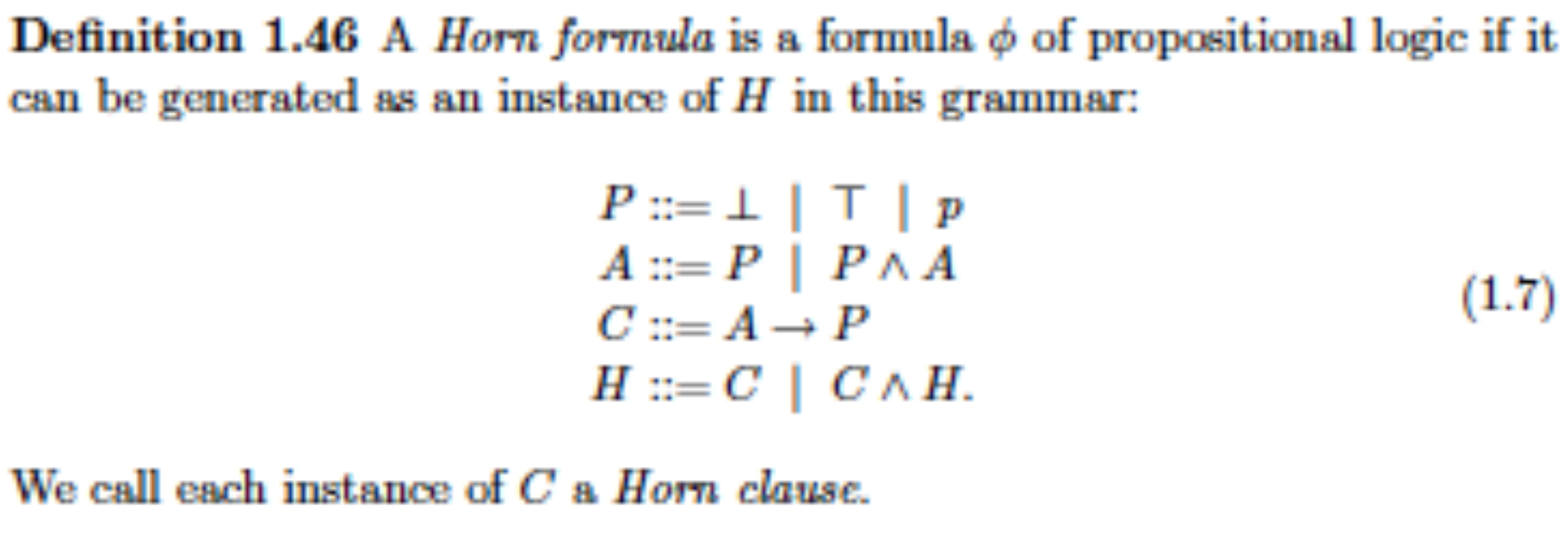
Compute CNF (NNF (IMPL_FREE $\neg(p \to (\neg(q \wedge (\neg p \to q))))))$

CNF

Validity is easy

Satisfiability is difficult

Particular sentences: Horn clauses

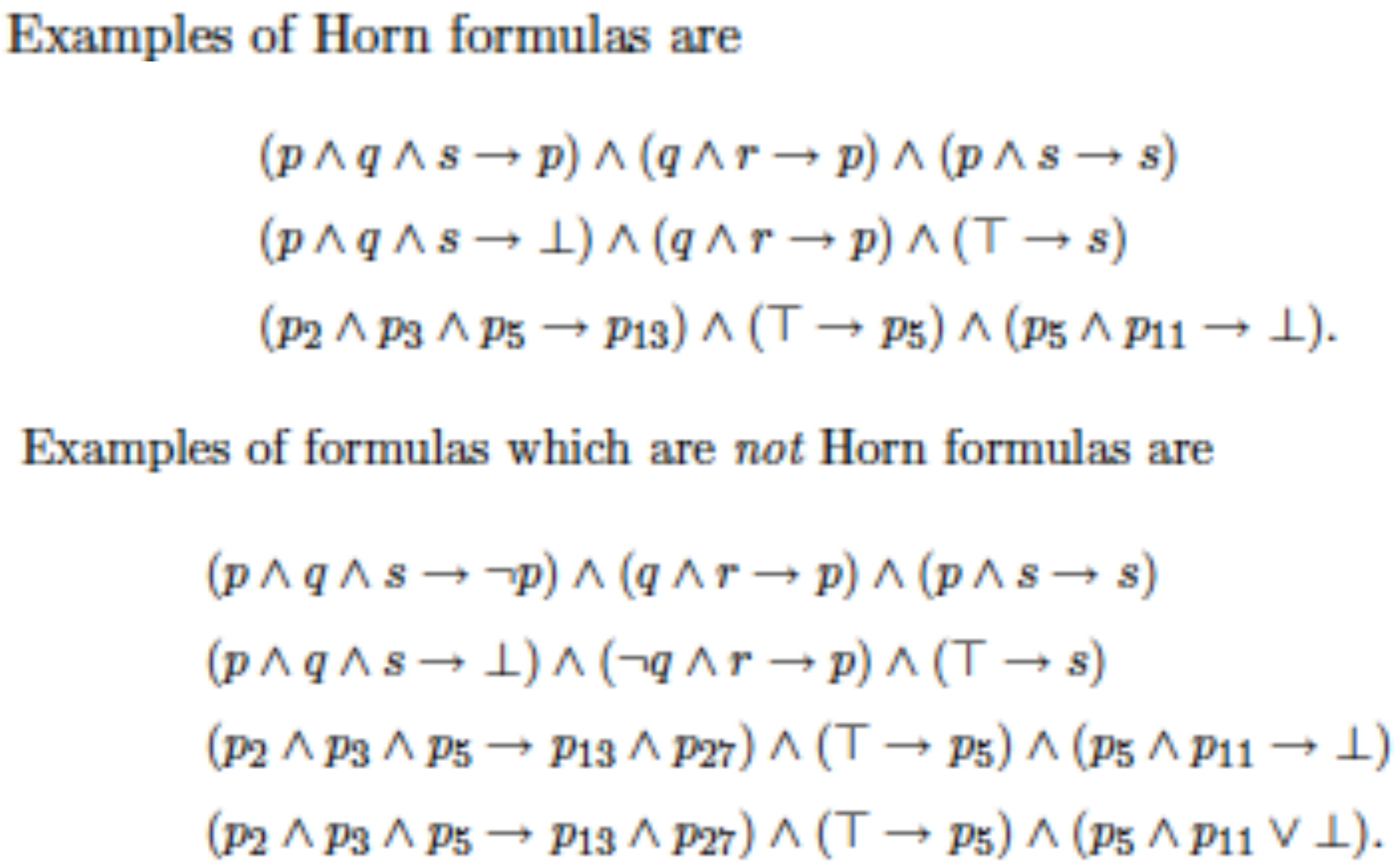


Definition 1.46 A *Horn formula* is a formula ϕ of propositional logic if it can be generated as an instance of H in this grammar:

$$\begin{aligned} P &::= \bot \mid \top \mid p \\ A &::= P \mid P \wedge A \\ C &::= A \to P \\ H &::= C \mid C \wedge H. \end{aligned} \tag{1.7}$$

We call each instance of C a *Horn clause*.

Particular sentences: Horn clauses



Examples of Horn formulas are

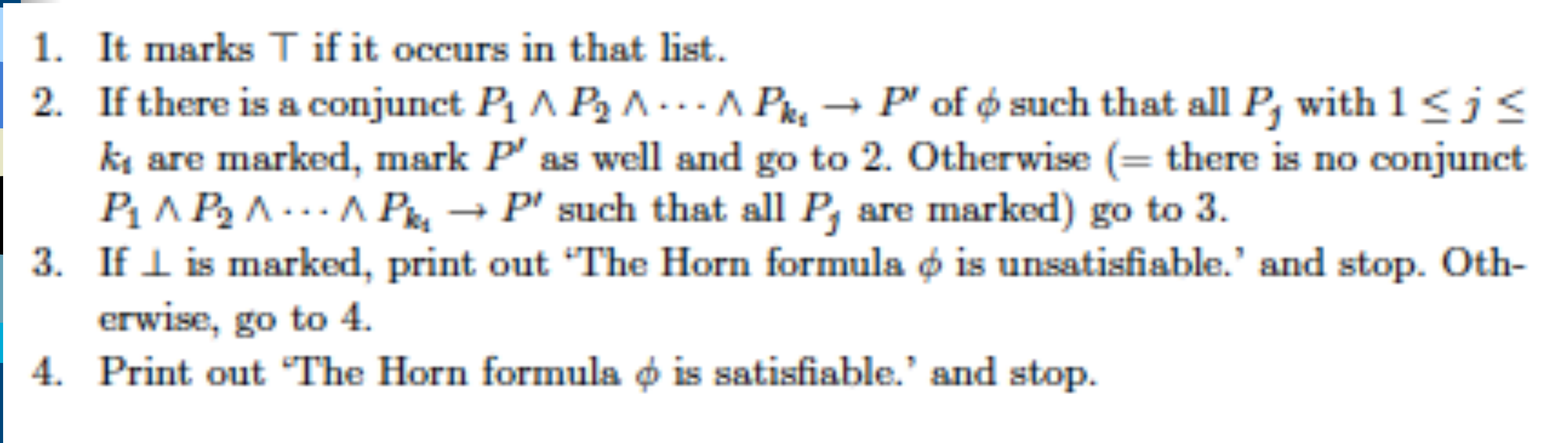
$$(p \wedge q \wedge s \to p) \wedge (q \wedge r \to p) \wedge (p \wedge s \to s)$$
$$(p \wedge q \wedge s \to \bot) \wedge (q \wedge r \to p) \wedge (\top \to s)$$
$$(p_2 \wedge p_3 \wedge p_5 \to p_{13}) \wedge (\top \to p_5) \wedge (p_5 \wedge p_{11} \to \bot).$$

Examples of formulas which are *not* Horn formulas are

$$(p \wedge q \wedge s \to \neg p) \wedge (q \wedge r \to p) \wedge (p \wedge s \to s)$$
$$(p \wedge q \wedge s \to \bot) \wedge (\neg q \wedge r \to p) \wedge (\top \to s)$$
$$(p_2 \wedge p_3 \wedge p_5 \to p_{13} \wedge p_{27}) \wedge (\top \to p_5) \wedge (p_5 \wedge p_{11} \to \bot)$$
$$(p_2 \wedge p_3 \wedge p_5 \to p_{13} \wedge p_{27}) \wedge (\top \to p_5) \wedge (p_5 \wedge p_{11} \vee \bot).$$

Horn clauses: satisfiability

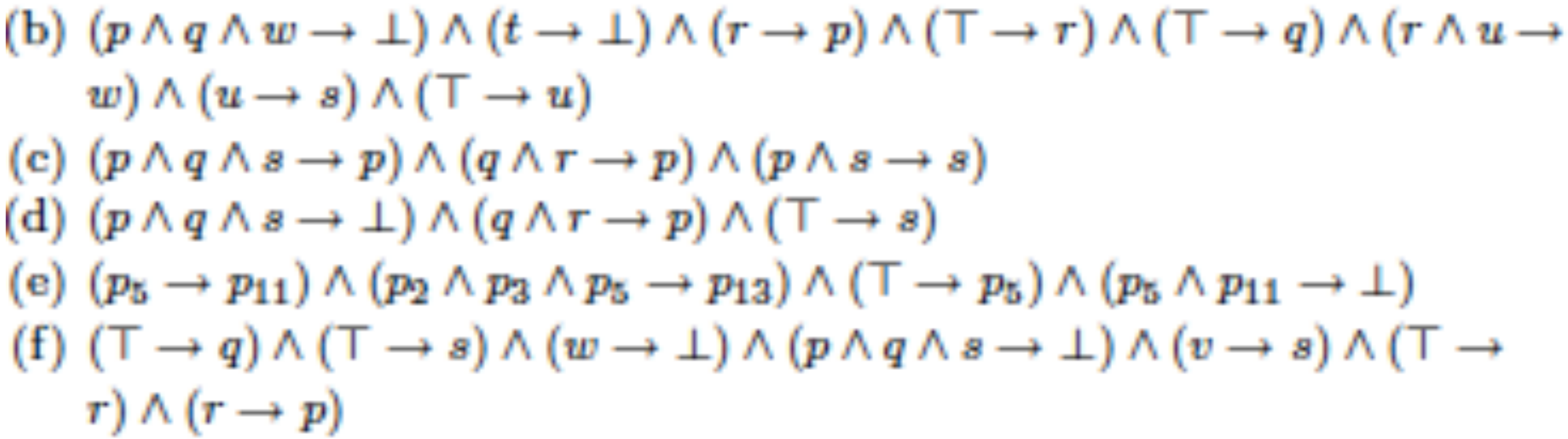
The algorithm



1. It marks $\top$ if it occurs in that list.
2. If there is a conjunct $P_1 \wedge P_2 \wedge \cdots \wedge P_{k_i} \to P'$ of ϕ such that all P_j with $1 \le j \le k_i$ are marked, mark P' as well and go to 2. Otherwise (= there is no conjunct $P_1 \wedge P_2 \wedge \cdots \wedge P_{k_i} \to P'$ such that all P_j are marked) go to 3.
3. If $\bot$ is marked, print out 'The Horn formula ϕ is unsatisfiable.' and stop. Otherwise, go to 4.
4. Print out 'The Horn formula ϕ is satisfiable.' and stop.

Horn clauses: satisfiability

Exercises



(b) $(p \wedge q \wedge w \rightarrow \bot) \wedge (t \rightarrow \bot) \wedge (r \rightarrow p) \wedge (\top \rightarrow r) \wedge (\top \rightarrow q) \wedge (r \wedge u \rightarrow w) \wedge (u \rightarrow s) \wedge (\top \rightarrow u)$

(c) $(p \wedge q \wedge s \rightarrow p) \wedge (q \wedge r \rightarrow p) \wedge (p \wedge s \rightarrow s)$

(d) $(p \wedge q \wedge s \rightarrow \bot) \wedge (q \wedge r \rightarrow p) \wedge (\top \rightarrow s)$

(e) $(p_5 \rightarrow p_{11}) \wedge (p_2 \wedge p_3 \wedge p_5 \rightarrow p_{13}) \wedge (\top \rightarrow p_5) \wedge (p_5 \wedge p_{11} \rightarrow \bot)$

(f) $(\top \rightarrow q) \wedge (\top \rightarrow s) \wedge (w \rightarrow \bot) \wedge (p \wedge q \wedge s \rightarrow \bot) \wedge (v \rightarrow s) \wedge (\top \rightarrow r) \wedge (r \rightarrow p)$

Exercises

1. Given the following formulas, draw their corresponding parse tree:

   (a) p

   * (b) $p \wedge q$

   (c) $p \wedge \neg q \to \neg p$

   * (d) $p \wedge (\neg q \to \neg p)$

   (e) $p \to (\neg q \vee (q \to p))$

   * (f) $\neg((\neg q \wedge (p \to r)) \wedge (r \to q))$

   (g) $\neg p \vee (p \to q)$

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

   (i) $((s \vee (\neg p)) \to (\neg p))$

   (j) $(s \vee ((\neg p) \to (\neg p)))$

   (k) $(((s \to (r \vee l)) \vee ((\neg q) \wedge r)) \to ((\neg(p \to s)) \to r))$

   (l) $(p \to q) \wedge (\neg r \to (q \vee (\neg p \wedge r)))$.

Exercises

These exercises let you practice proofs using mathematical induction. Make sure that you state your base case and inductive step clearly. You should also indicate where you apply the induction hypothesis.

(a) Prove that

$$(2 \cdot 1 - 1) + (2 \cdot 2 - 1) + (2 \cdot 3 - 1) + \cdots + (2 \cdot n - 1) = n^2$$

by mathematical induction on $n \geq 1$.

(b) Let k and l be natural numbers. We say that k is divisible by l if there exists a natural number p such that $k = p \cdot l$. For example, 15 is divisible by 3 because $15 = 5 \cdot 3$. Use mathematical induction to show that $11^n - 4^n$ is divisible by 7 for all natural numbers $n \geq 1$.

(c) Use mathematical induction to show that

$$1^2 + 2^2 + 3^2 + \cdots + n^2 = \frac{n \cdot (n+1) \cdot (2n+1)}{6}$$

for all natural numbers $n \geq 1$.

(d) Prove that $2^n \geq n + 12$ for all natural numbers $n \geq 4$. Here the base case is $n = 4$. Is the statement true for any $n < 4$?