

# 形式化方法导引

## 第 4 章 逻辑问题求解

### 4.2 理论 - (3) CNF 与 Horn Clauses

黄文超

<https://faculty.ustc.edu.cn/huangwenchao>

→ 教学课程 → 形式化方法导引

## 2. 理论

### 2.3 CNF and Horn Clauses | 回顾

回顾: SAT 求解所遇到的问题:

Provable equivalence:

$$\begin{array}{ll}\neg(p \wedge q) \dashv\vdash \neg q \vee \neg p & \neg(p \vee q) \dashv\vdash \neg p \wedge \neg q \\ p \rightarrow q \dashv\vdash \neg q \rightarrow \neg p & p \rightarrow q \dashv\vdash \neg p \vee q \\ p \wedge q \rightarrow r \dashv\vdash r \vee \neg p \vee \neg q & p \wedge q \rightarrow r \dashv\vdash p \rightarrow (q \rightarrow r).\end{array}$$

rules 太多: 推演过于复杂, 符号也有冗余

- 减少冗余的符号, 设计自动推演算法

问题: 如何减少冗余的符号, 设计自动推演算法?

先给部分结果:

- CNF (conjunctive normal form) 合取范式
  - 取如下 (一元、二元) 符号
    - $\{\wedge, \vee, \neg\}$
- Horn clauses 霍恩子句
  - 取如下 (一元、二元) 符号
    - $\{\wedge, \rightarrow\}$

## 2. 理论

### 2.3 CNF and Horn Clauses | 回顾

回顾: SAT 的一种求解思路:

Two Problems:

- Problem 1: Checking SAT of a proposition formula
- Problem 2: Checking SAT of a CNF formula

How to solve problem 1?

- Step 1: Transform Problem 1 to Problem 2
- Step 2: Solve Problem 2.

Step 1 (one way by applying the following rules):

- $\neg, \vee, \wedge$ : Do nothing
- $\rightarrow$ :  $p \rightarrow q \equiv \neg p \vee q$
- $\leftrightarrow$ :  $p \leftrightarrow q \equiv (p \rightarrow q) \wedge (q \rightarrow p)$

Step 1 (another clever way): *Tseitin transformation*.

Step 2: 已解

## 2. 理论

### 2.3 CNF and Horn Clauses | 本节内容

#### 本节内容:

How to transform a propositional formula to CNF?

- *challenge*:
  - show how it is possible
  - why a naive solution may blow up
- *Tseitin transformation*
  - linear in the size of the formula
  - used in current SAT solvers

How to solve SAT based on *Horn clauses* instead of CNF?

## 2. 理论

### 2.3 CNF and Horn Clauses | Transform a propositional formula to CNF | Challenges

For any formula  $\phi$  we can make its truth table

For *any* 0 in this truth table, we can make a corresponding clause

$p$	$q$	$r$	$\phi$	$p \vee \neg q \vee r$	$\neg p \vee q \vee \neg r$	$\neg p \vee \neg q \vee \neg r$
0	0	0	1	1	1	1
0	0	1	1	1	1	1
0	1	0	0	0	1	1
0	1	1	1	1	1	1
1	0	0	1	1	1	1
1	0	1	0	1	0	1
1	1	0	1	1	1	1
1	1	1	0	1	1	0

Now the conjunction  $(p \vee \neg q \vee r) \wedge (\neg p \vee q \vee \neg r) \wedge (\neg p \vee \neg q \vee \neg r)$  of these clauses has the same truth table as  $\phi$ , so it is logically equivalent to  $\phi$

## 2. 理论

### 2.3 CNF and Horn Clauses | Transform a propositional formula to CNF | Challenges

This approach always works: if the truth table of  $\phi$  *contains  $k$  0's*, then we obtain a CNF consisting of  *$k$  clauses*

*Drawback:* this  $k$  may be very large

- consider the case:  $n$  variables in  $\phi$ 
  - How many clauses for constructing  $\phi$ ?
  - How many literals for each clause?

*Good case:* A smaller CNF logically equivalent to  $\phi$  may exist, have clauses of  $\leq n$  literals

- Example:  $p \wedge (\neg q \vee r)$  is a CNF with 2 clauses, having 5 0's in truth table of 8 rows

*Bad case:* For some formulas the *exponential* number of clauses is *unavoidable* (见下页)

# 形式化方法导引

## └ 2. 理论

### 2. 理论

2.1 CNF and Horn Clauses | Transform a propositional formula to CNF | Challenges

This approach always works: if the truth table of  $\phi$  contains  $k$  0's, then we obtain a CNF consisting of  $k$  clauses

**Drawback:** this  $k$  may be very large

- ◆ consider the case:  $n$  variables in  $\phi$ 
  - How many clauses for constructing  $\phi$ ?
  - How many literals for each clause?

**Good case:** A smaller CNF logically equivalent to  $\phi$  may exist, have clauses of  $\leq n$  literals

- ◆ Example:  $p \wedge (\neg q \vee r)$  is a CNF with 2 clauses, having 5 0's in truth table of 8 rows

**Bad case:** For some formulas the *exponential* number of clauses is *unavoidable* (见下页)

Drawback: if there are  $n$  variables, then the truth table has  $2^n$  rows:

*exponential* in  $n$

All of these clauses have exactly  $n$  literals

## 2. 理论

### 2.3 CNF and Horn Clauses | Transform a propositional formula to CNF | Challenges

**Example:**  $\Phi : (\cdots ((p_1 \leftrightarrow p_2) \leftrightarrow p_3) \cdots \leftrightarrow p_n)$

This formula yields true iff an *even number* of  $p_i$ 's has the value false

#### 命题

Let  $X$  be a CNF satisfying  $\Phi \equiv X$

Then every clause  $C$  in  $X$  contains exactly  $n$  literals

证明:

Assume not, then some  $p_i$  does not occur in a clause  $C$  of  $X$

Then you can give values to the remaining variables such that  $C$  is false, and  $X$  is false too, independent of the value of  $p_i$

Swapping values of  $p_i$  does not swap values of  $X$ , contradicting  $\Phi \equiv X$

□

## 2. 理论

### 2.3 CNF and Horn Clauses | Transform a propositional formula to CNF | Challenges

**Example:**  $\Phi : (\dots ((p_1 \leftrightarrow p_2) \leftrightarrow p_3) \dots \leftrightarrow p_n)$

#### 命题

Let  $X$  be a CNF satisfying  $\Phi \equiv X$

Then every clause  $C$  in  $X$  contains exactly  $n$  literals

The truth table of  $\Phi$  contains  $2^n$  rows, half of which containing 0

So exactly  $2^{n-1}$  rows contain 0

Every clause of exactly  $n$  literals has one 0 in its truth table

So we need  $2^{n-1}$  *such clauses* to obtain the truth table of  $\Phi$

So for this  $\Phi$  the *exponential* size is unavoidable

## 2. 理论

### 2.3 CNF and Horn Clauses | Transform a propositional formula to CNF | Challenges

Summarizing the *challenge*:

- For *any* propositional formula  $\phi$ , it is possible to find a logically equivalent CNF
- Bad case: but the size of this CNF may be *exponential*

新方法: *Tseitin transformation* (见下页)

## 2. 理论

### 2.3 CNF and Horn Clauses | Tseitin transformation

#### *Tseitin transformation*

- *Linear* transformation of *arbitrary* propositional formula to CNF

思路: Give a name to every subformula (except literals) and use this name as a *fresh* variable

- For every formula  $\phi$  on  $\leq 3$  variables there is a small CNF  $\text{cnf}(\phi) \equiv \phi$
- Transform a *big* formula  $\phi$  to the conjunction of  $\text{cnf}(\phi_i)$  for many small formulas  $\phi_i$  obtained from  $\phi$ , one for each subformula

More precisely, for every subformula  $\psi$ , we define

- $n_\psi = \psi$ , if  $\psi$  is a literal
- $n_\psi$  = the name of  $\psi$ , otherwise

# 形式化方法导引

## └ 2. 理论

参考论文: GS Tseitin, On the complexity of derivation in propositional calculus, 引用次数:1973

### 2. 理论

2.1 CNF and Horn Clauses | Tseitin transformation

#### Tseitin transformation

- Linear transformation of *arbitrary* propositional formula to CNF

思路: Give a name to every subformula (except literals) and use this name as a *fresh* variable

- For every formula  $\phi$  on  $\leq 3$  variables there is a small CNF  $\text{cnf}(\phi) \equiv \phi$
- Transform a *big* formula  $\phi$  to the conjunction of  $\text{cnf}(\phi_i)$  for many small formulas  $\phi_i$  obtained from  $\phi$ , one for each subformula

More precisely, for every subformula  $\psi$ , we define

- $n_\psi = \psi$ , if  $\psi$  is a literal
- $n_\psi$  = the name of  $\psi$ , otherwise

## 2. 理论

### 2.3 CNF and Horn Clauses | Tseitin transformation

More precisely, for every subformula  $\psi$

- $n_\psi = \psi$ , if  $\psi$  is a literal
- $n_\psi$  = the name of  $\psi$ , otherwise

The Tseitin transformation  $T(\phi)$  of  $\phi$ , is defined to be the CNF consisting of:

- $n_\psi$
- $\text{cnf}(q \leftrightarrow \neg n_\psi)$  for every non-literal subformula of the shape  $\neg\psi$  having name  $q$
- $\text{cnf}(q \leftrightarrow (n_{\psi_1} \diamond n_{\psi_2}))$  for every subformula of the shape  $\psi_1 \diamond \psi_2$  having name  $q$ , for

$$\diamond \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$$

Example  $\phi$ :

$$\underbrace{(\neg s \wedge p)}_B \leftrightarrow \underbrace{((\underbrace{q \rightarrow r}_D) \vee \neg p)}_C$$

yields  $T(\phi)$ :

$$\begin{aligned} & n_\phi \wedge \\ & \text{cnf}(n_\phi \leftrightarrow (B \leftrightarrow C)) \wedge \\ & \text{cnf}(B \leftrightarrow (\neg s \wedge p)) \wedge \\ & \text{cnf}(C \leftrightarrow (D \vee \neg p)) \wedge \\ & \text{cnf}(D \leftrightarrow (q \rightarrow r)) \end{aligned}$$

## 2. 理论

### 2.3 CNF and Horn Clauses | Tseitin transformation | Preservation of satisfiability

#### 定理

$\phi$  is satisfiable if and only if  $T(\phi)$  is satisfiable

证明: 略 (若感兴趣, 可见 note)

剩下的问题: We still need to compute the formula  $cnf(n_\psi \leftrightarrow \dots)$

$$\begin{aligned} cnf(p \leftrightarrow \neg q) &= (p \vee q) \\ &\quad \wedge (\neg p \vee \neg q) \end{aligned}$$

$$\begin{aligned} cnf(p \leftrightarrow (q \vee r)) &= (\neg p \vee q \vee r) \\ &\quad \wedge (p \vee \neg q) \\ &\quad \wedge (p \vee \neg r) \end{aligned}$$

$$\begin{aligned} cnf(p \leftrightarrow (q \wedge r)) &= (p \vee \neg q \vee \neg r) \\ &\quad \wedge (\neg p \vee q) \\ &\quad \wedge (\neg p \vee r) \end{aligned}$$

$$\begin{aligned} cnf(p \leftrightarrow (q \leftrightarrow r)) &= (p \vee q \vee r) \\ &\quad \wedge (p \vee \neg q \vee \neg r) \\ &\quad \wedge (\neg p \vee q \vee \neg r) \\ &\quad \wedge (\neg p \vee \neg q \vee r) \end{aligned}$$

## └ 2. 理论

## 定理

 $\phi$  is satisfiable if and only if  $T(\phi)$  is satisfiable

证明: 略 (若感兴趣, 可见 note)

剩下的问题: We still need to compute the formula  $cnf(n_{i_0} \leftrightarrow \dots)$ 

$cnf(p \leftrightarrow \neg q) = (p \vee q)$	$cnf(p \leftrightarrow (q \vee r)) = (\neg p \vee q \vee r)$
$\wedge (\neg p \vee \neg q)$	$\wedge (p \vee \neg q)$
	$\wedge (p \vee \neg r)$
$cnf(p \leftrightarrow (q \wedge r)) = (p \vee \neg q \vee \neg r)$	$cnf(p \leftrightarrow (q \leftrightarrow r)) = (p \vee q \vee r)$
$\wedge (\neg p \vee q)$	$\wedge (p \vee \neg q \vee \neg r)$
$\wedge (\neg p \vee r)$	$\wedge (\neg p \vee q \vee \neg r)$
	$\wedge (\neg p \vee \neg q \vee r)$

证: (1) let  $\phi$  is satisfiable, then it admits a satisfying assignment.

Extend this to  $n_\psi$  for subformula  $\psi$ :  $n_\psi$  gets the value of the subformula  $\psi$

Then by construction this yields a satisfying assignment for  $T(\phi)$ :

- $n_\phi$  yields true
- $q \leftrightarrow \neg n_\psi$  yields true for subformula  $\neg\psi$  with name  $q$ , so does  $cnf(q \leftrightarrow \neg n_\psi)$
- $q \leftrightarrow (n_{\psi_1} \diamond n_{\psi_2})$  yields true for subformula  $\psi_1 \diamond \psi_2$  having name  $q$ , for  $\diamond \in \{\vee, \wedge, \rightarrow, \leftrightarrow\}$ , so does  $cnf(q \leftrightarrow (n_{\psi_1} \diamond n_{\psi_2}))$

So satisfiability of  $\phi$  implies satisfiability of  $T(\phi)$

## └ 2. 理论

## 定理

 $\phi$  is satisfiable if and only if  $T(\phi)$  is satisfiable

证明: 略 (若感兴趣, 可见 note)

剩下的问题: We still need to compute the formula  $\text{cnf}(n_{\phi_0} \leftrightarrow \dots)$ 

$\text{cnf}(p \leftrightarrow \neg q) = (p \vee q)$	$\text{cnf}(p \leftrightarrow (q \vee r)) = (\neg p \vee q \vee r)$
$\wedge (\neg p \vee \neg q)$	$\wedge (p \vee \neg q)$
	$\wedge (p \vee \neg r)$
$\text{cnf}(p \leftrightarrow (q \wedge r)) = (p \vee \neg q \vee \neg r)$	$\text{cnf}(p \leftrightarrow (q \leftrightarrow r)) = (p \vee q \vee r)$
$\wedge (\neg p \vee q)$	$\wedge (p \vee \neg q \vee \neg r)$
$\wedge (\neg p \vee r)$	$\wedge (\neg p \vee q \vee \neg r)$
	$\wedge (\neg p \vee \neg q \vee r)$

(2) Conversely, assume  $T(\phi)$  is satisfiable = admits a satisfying assignment

Apply this same satisfying assignment to the original formula  $\phi$

Since  $q \leftrightarrow (n_{\psi_1} \diamond n_{\psi_2})$  yields true for every subformula  $\psi_1 \diamond \psi_2$  having name  $q$  (and similar for  $\neg n_{\psi}$ ), we obtain that every subformula  $\psi$  of  $\phi$  gets the value of  $n_{\psi}$

Since  $n_{\phi}$  yields true (as part of  $T(\phi)$ ), we obtain that the original formula  $\phi$  yields true, so  $\phi$  is satisfiable



## 2. 理论

### 2.3 CNF and Horn Clauses | Tseitin transformation | Preservation of satisfiability

#### Concluding

- For every propositional formula  $\phi$  its Tseitin transformation  $T(\phi)$  is easily computed
- Size of  $T(\phi)$  is linear in size of  $\phi$
- Preserves satisfiability
- Not only CNF, even *3-CNF*
- Used in current SAT solvers

## 2. 理论

### 2.3 CNF and Horn Clauses | 本节内容

回顾本节内容:

How to transform a propositional formula to CNF?

- *challenge*:
  - show how it is possible
  - why a naive solution may blow up
- *Tseitin transformation*
  - linear in the size of the formula
  - used in current SAT solvers

问题: How to solve SAT based on *Horn clauses* instead of CNF?

## 2. 理论

### 2.3 CNF and Horn Clauses | Horn clauses

#### 定义: Horn clause

A *Horn formula* is a formula  $\phi$  propositional logic if it can be generated as instance of  $H$  in this grammar:

$$P ::= \perp \mid \top \mid p$$

$$A ::= P \mid P \wedge A$$

$$C ::= A \rightarrow P$$

$$H ::= C \mid C \wedge H$$

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

Recall that the logical constants:

- $\perp$  denotes an unsatisfiable formula
- $\top$  denotes a tautology

例: Examples of *Horn formulas*:

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

## 2. 理论

### 2.3 CNF and Horn Clauses | Horn clauses

#### 算法: HORN( $\phi$ )

##### begin function

mark all occurrences of  $\top$  in  $\phi$ ;

**while** there is a conjunct  $P_1 \wedge P_2 \wedge \cdots \wedge P_{k_i} \rightarrow P'$  of  $\phi$  such that  
all  $P_j$  are marked but  $P'$  isn't

**do** mark  $P'$

**end while**

**if**  $\perp$  is marked

**then return** 'unsatisfiable'

**else return** 'satisfiable'

**end function**

## 2. 理论

### 2.3 CNF and Horn Clauses | Horn clauses

#### 例 1: Horn ( $\phi$ )

$$\phi = (p \wedge q \wedge w \rightarrow \perp) \wedge (t \rightarrow \perp) \wedge (r \rightarrow p) \wedge (\top \rightarrow r) \wedge (\top \rightarrow q) \wedge (u \rightarrow s) \wedge (\top \rightarrow u)$$

Marked:  $\top$   $r$   $q$   $u$   $p$   $s$  return 'satisfiable'

#### 例 2: Horn ( $\phi$ )

$$\phi = (p \wedge q \wedge w \rightarrow \perp) \wedge (t \rightarrow \perp) \wedge (r \rightarrow p) \wedge (\top \rightarrow r) \wedge (\top \rightarrow q) \wedge (r \wedge u \rightarrow w) \wedge (u \rightarrow s) \wedge (\top \rightarrow u)$$

Marked:  $\top$   $r$   $q$   $u$   $p$   $w$   $\perp$  return 'unsatisfiable'

#### 例 3: Horn ( $\phi$ )

$$\phi = (p \wedge q \wedge s \rightarrow p) \wedge (q \wedge r \rightarrow p) \wedge (p \wedge s \rightarrow s)$$

Marked: None... return 'satisfiable'

## 2. 理论

### 2.3 CNF and Horn Clauses | Horn clauses

#### Concluding

- There are practically important subclasses, e.g., *Horn formulas*, which have much more *efficient ways* of deciding their satisfiability
- *Horn clauses* have been applied to many classical formal verifiers, e.g., the protocol verifier ProVerif.
- How to transform propositional formulas to Horn formulas?

# 作业

1. Construct a formula in CNF based on the following truth table:

$p$	$q$	$r$	$\phi$
1	1	1	0
1	1	0	1
1	0	1	0
0	1	1	1
1	0	0	0
0	1	0	0
0	0	1	1
0	0	0	0

2. Apply algorithm HORN to each of these Horn formulas:

- ①  $(p \wedge q \wedge s \rightarrow \perp) \wedge (q \wedge r \rightarrow p) \wedge (\top \rightarrow s)$
- ②  $(p_5 \rightarrow p_{11}) \wedge (p_2 \wedge p_5 \rightarrow p_{13}) \wedge (\top \rightarrow p_5) \wedge (p_5 \wedge p_{11} \rightarrow \perp)$
- ③  $(\top \rightarrow q) \wedge (\top \rightarrow s) \wedge (w \rightarrow \perp) \wedge (p \wedge q \wedge s \rightarrow \perp) \wedge (v \rightarrow s) \wedge (\top \rightarrow s) \wedge (r \rightarrow p)$