# Theory of Computation
# Chapter 4

Guan-Shieng Huang

Apr. 7, 2003
Feb. 19, 2006

# Boolean Expressions

- $X = \{x_1, x_2, \ldots\}$ a countably infinite variables, each can be TRUE or FALSE.
- Logical connectivities:

  $\vee$: logical or; $\wedge$: logical and; $\neg$: logical not.
- The syntax:

  A Boolean expression can be one of

  1. a Boolean variable, such as $x_i$;
  2. $\neg \phi_1$;
  3. $(\phi_1 \vee \phi_2)$;
  4. $(\phi_1 \wedge \phi_2)$;

  where $\phi_1, \phi_2$ are Boolean expressions. (Inductive definition)

# Remarks

- $\neg\phi_1$: the negation of $\phi_1$
- $(\phi_1 \vee \phi_2)$: the disjunction of $\phi_1$ and $\phi_2$
- $(\phi_1 \wedge \phi_2)$: the conjunction of $\phi_1$ and $\phi_2$
- $x_i$ or $\neg x_i$ is called a literal.

# The Semantics

A truth assignment $T$ is a mapping from a set of variables $X' \subset X$ to $\{\textsc{True}, \textsc{false}\}$.

1. $T \models x_i$ if $T(x_i) = \textsc{true}$;
2. $T \models \neg\phi$ if not $T \models \phi$;
3. $T \models (\phi_1 \vee \phi_2)$ if $T \models \phi_1$ or $T \models \phi_2$;
4. $T \models (\phi_1 \wedge \phi_2)$ if $T \models \phi_1$ and $T \models \phi_2$;

where $T$ is appropriate.

# Example

$\phi = ((\neg x_1 \vee x_2) \wedge x_3)$ and

$T = \{x_1 \rightarrow \text{TRUE}, x_2 \rightarrow \text{FALSE}, x_3 \rightarrow \text{TRUE}\}$,

then $T \not\models \phi$.

$\because T \not\models \neg x_1$ and $T \not\models x_2$, $\therefore T \not\models (\neg x_1 \vee x_2)$.

# Remark

1. $(\phi_1 \Rightarrow \phi_2)$ as a shorthand of $(\neg\phi_1 \vee \phi_2)$.
2. $(\phi_1 \Leftrightarrow \phi_2)$ as a shorthand of $((\phi_1 \Rightarrow \phi_2) \wedge (\phi_2 \Rightarrow \phi_1))$.

Two expressions $\phi_1$ and $\phi_2$ are equivalent if $T \models \phi_1$ if and only if $T \models \phi_2$ for all appropriate $T$. Written as $\phi_1 \equiv \phi_2$.

1. $(\phi_1 \vee \phi_2) \equiv (\phi_2 \vee \phi_1)$; (commutative law)
2. $(\phi_1 \wedge \phi_2) \equiv (\phi_2 \wedge \phi_1)$;
3. $\neg\neg\phi_1 \equiv \phi_1$; (double-negation law)
4. $((\phi_1 \vee \phi_2) \vee \phi_3) \equiv (\phi_1 \vee (\phi_2 \vee \phi_3))$; (associative law)
5. $((\phi_1 \wedge \phi_2) \wedge \phi_3) \equiv (\phi_1 \wedge (\phi_2 \wedge \phi_3))$;
6. $((\phi_1 \wedge \phi_2) \vee \phi_3) \equiv ((\phi_1 \vee \phi_3) \wedge (\phi_2 \vee \phi_3))$; (distributive law)
7. $((\phi_1 \vee \phi_2) \wedge \phi_3) \equiv ((\phi_1 \wedge \phi_3) \vee (\phi_2 \wedge \phi_3))$;
8. $\neg(\phi_1 \vee \phi_2) \equiv (\neg\phi_1 \wedge \neg\phi_2)$; (De Morgan's law)
9. $\neg(\phi_1 \wedge \phi_2) \equiv (\neg\phi_1 \vee \neg\phi_2)$;
10. $(\phi_1 \vee \phi_1) \equiv \phi_1$. (idempotent law)

And $\wedge$ and $\vee$ are dual. You can interchange all $\wedge$'s with all $\vee$'s.

## Remarks

1. $\bigwedge_{i=1}^{n} \phi_i$ stands for $(\phi_1 \wedge \phi_2 \wedge \cdots \wedge \phi_n)$.
2. $\bigvee_{i=1}^{n} \phi_i$ stands for $(\phi_1 \vee \phi_2 \vee \cdots \vee \phi_n)$.

# Normal Forms

**Conjunctive-normal form:**
$\phi = \bigwedge_{i=1}^{n} C_i$ where $n \geq 1$ and each $C_i$ is the disjunction of literals.
$C_i$ is called a clause.

$$(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2)$$

**Disjunctive-normal form:**
$\phi = \bigvee_{i=1}^{n} D_i$ where $n \geq 1$ and each $D_i$ is the conjunction of literals.
$D_i$ is called an implicant.

$$(x_1 \wedge \neg x_2) \vee (\neg x_1 \wedge x_2)$$

# Theorem 4.1

Every Boolean expression is equivalent to one in CNF (also one in DNF).

**Example**

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

# Satisfiability

- A Boolean expression $\phi$ is satisfiable if there is a truth assignment $T$ such that $T \models \phi$.
- An expression $\phi$ is valid (or tautology) if $T \models \phi$ for all $T$ appropriate to $\phi$. (Written as $\models \phi$)
- $\phi$ is unsatisfiable if $T \not\models \phi$ for all $T$.

**Proposition 4.2**

A Boolean expression is unsatisfiable if and only if its negation is valid. ($\phi$ is unsatisfiable $\Longleftrightarrow \models \neg\phi$)

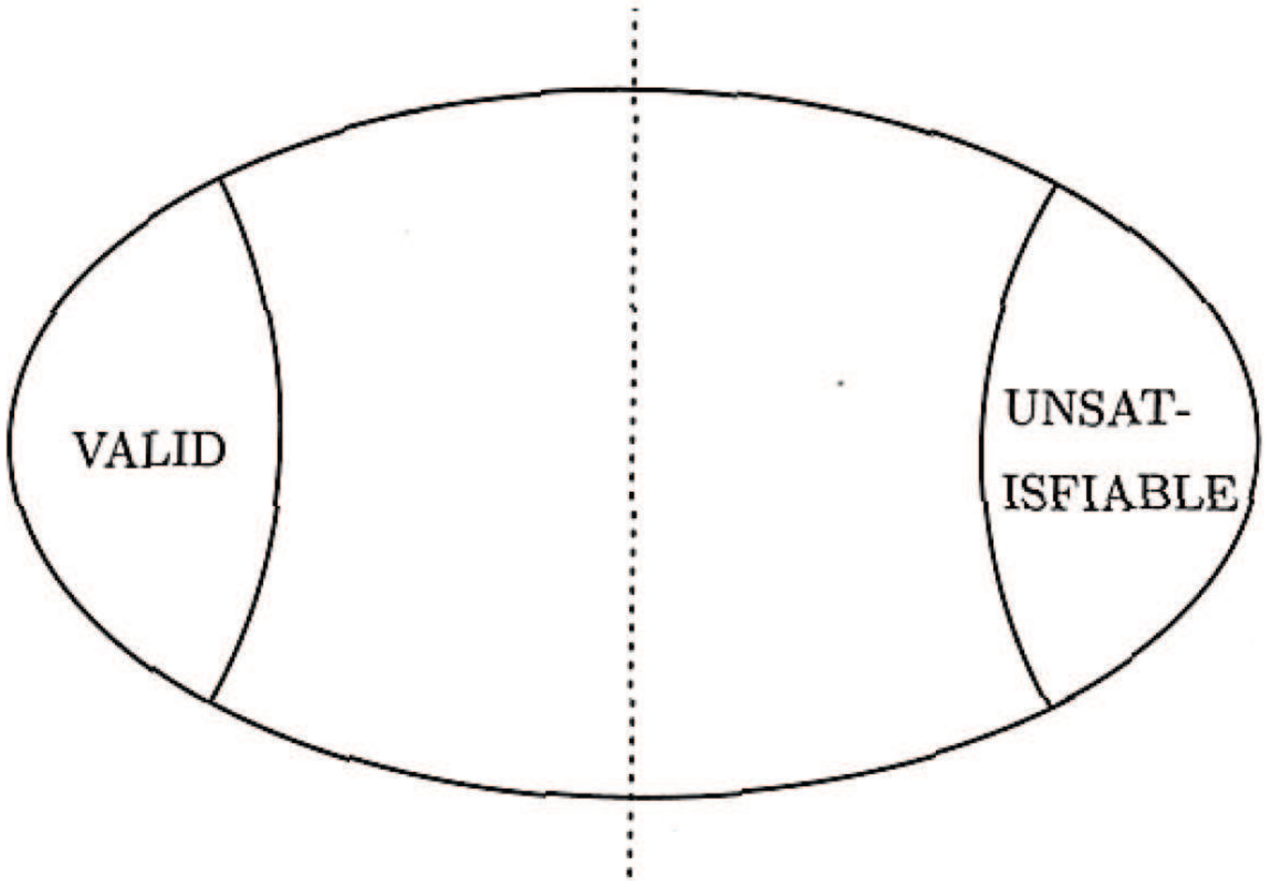


**Figure 4-1.** The geography of all Boolean expressions.

**Example 4.2**

1. $(x_1 \vee \neg x_2) \wedge \neg x_1$; (satisfiable)

2. $(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3) \wedge (x_3 \vee \neg x_1) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3)$. (unsatisfiable)

# SAT

Given any Boolean expression $\phi$ in conjunctive normal form, is it satisfiable?

**Remarks**

1. SAT $\in \mathcal{NP}$.
   (a) Guess an assignment.
   (b) Verify it.
2. SAT is NP-complete (Chap. 8).
3. SAT can be easily solved if $\phi$ is expressed in disjunctive normal form.
   $((\neg p \wedge \neg q) \vee (p \wedge q))$

# Horn

**Horn clause:** A clause is Horn if it has at most one positive literal.
$x_1 \wedge x_2 \cdots \wedge x_m \Rightarrow y$ could be written as $\neg x_1 \vee \neg x_2 \vee \cdots \vee \neg x_m \vee y$.

**Horn SAT:** Given any expression in the conjunction of Horn clauses, is it satisfiable?

**Example:**

$x_1 \vee \neg x_2, \quad x_1 \vee \neg x_3, \quad \neg x_2 \vee \neg x_3, \quad \neg x_1 \vee x_4, \quad x_1.$

$x_1 \Rightarrow x_2, \quad x_3 \Rightarrow x_1, \quad x_2 \wedge x_3 \Rightarrow \text{FALSE}, \quad x_1 \Rightarrow x_4, \quad x_1.$

**Algorithm**

1. Initially, $T := \emptyset$. (That is, all variables are set FALSE.)
2. Pick any unsatisfiable implication $x_1 \wedge x_2 \wedge \cdots \wedge x_m \Rightarrow y$ and add $y$ to $T$; repeat this rule until all implications are satisfied.

**Intuition:** Try to assign all variables on the premises false.

**Proposition:** Any assignment $T'$ satisfying $\phi$ must contain $T$. That is, $T$ is the minimum assignment satisfying $\phi$.

**Theorem 4.2** HORNSAT is in $\mathcal{P}$.

# Boolean Function

1. An $n$-ary Boolean function is a function from $\{\text{TRUE}, \text{FALSE}\}^n \to \{\text{TRUE}, \text{FALSE}\}$.
2. A Boolean expression $\phi$ expresses a Boolean function $f$ if for all truth value $t = (t_1, \ldots, t_n)$,

$$f(t) = \text{TRUE iff } T \models \phi,$$

where $T(x_i) = t_i$ for $1 \leq i \leq n$.

# Proposition 4.3

Any $n$-ary Boolean function $f$ can be expressed as a Boolean expression $\phi_f$ involving variables $x_1, x_2, \ldots, x_n$.

| $x_1$ | $x_2$ | $x_3$ | f |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

$$(\neg x_1 \wedge \neg x_2 \wedge \neg x_3) \vee (\neg x_1 \wedge x_2 \wedge \neg x_3) \vee (x_1 \wedge \neg x_2 \wedge x_3)$$

# Boolean Circuit

1. no cycle in the graph;
2. the in-degree of each node equals to 0, 1, or 2;
3. each node represents either TRUE, FALSE, $\wedge$, $\vee$, $\neg$, or a variable $x_i$.

$$(x_3 \wedge \neg((x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2))) \vee (\neg x_3 \wedge (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_2))$$

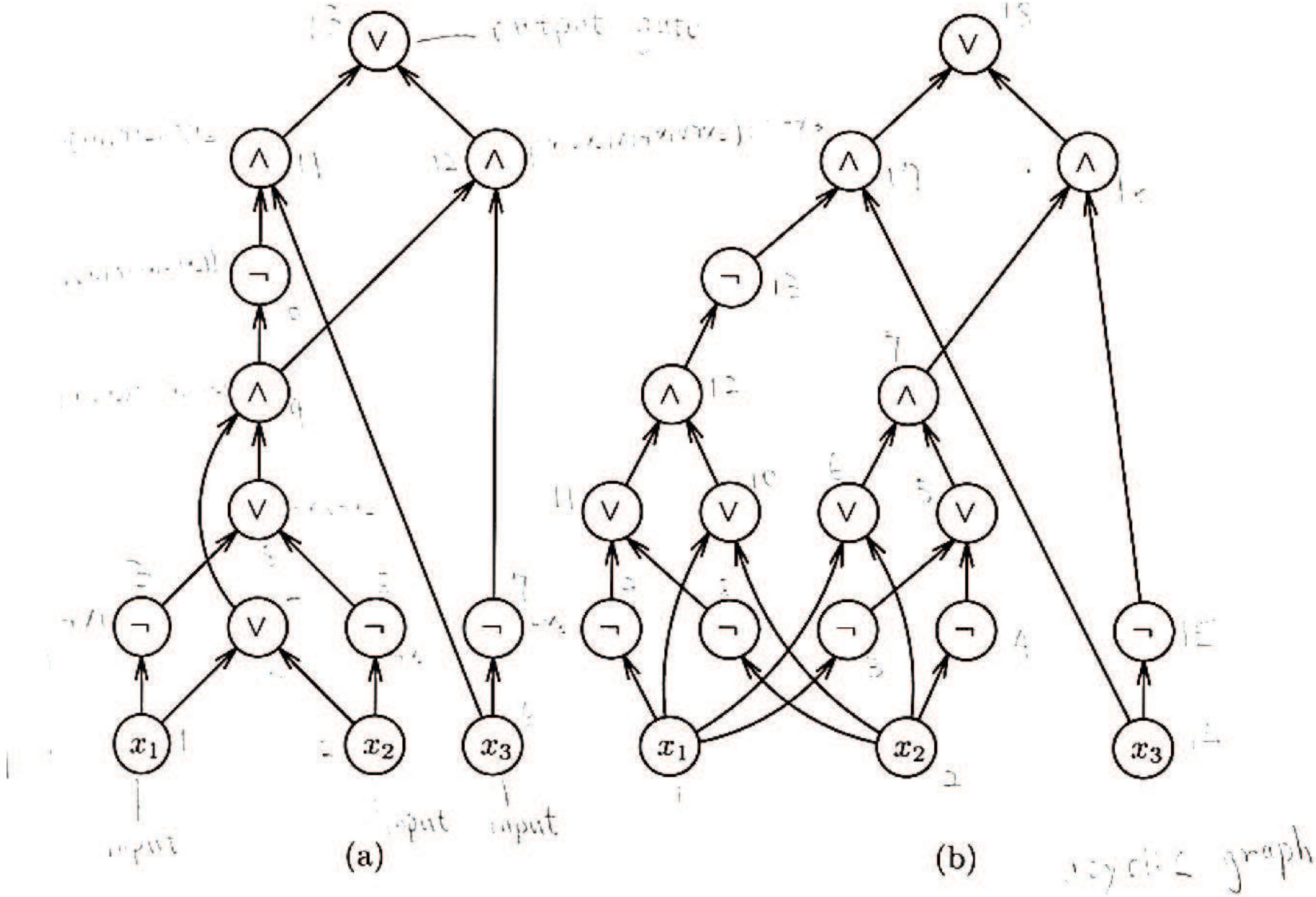


**Figure 4-2.** Two circuits.

**CIRCUIT SAT** Given any circuit $C$, is there a truth assignment $T$ appropriate to $C$ such that $T(C) = \text{TRUE}$?

**CIRCUIT VALUE** When an assignment $T$ is given, ask whether $T(C)$ is TRUE.

# Theorem 4.3

For any $n \geq 2$, there is an $n$-ary Boolean function $f$ such that no Boolean circuit with $\frac{2^n}{2n}$ or fewer gates can compute it.