Satisfiability Checking

03 Propositional logic II

Prof. Dr. Erika Ábrahám

RWTH Aachen University
Informatik 2
LuFG Theory of Hybrid Systems

WS 23/24

Literals

literal 为 或正或负的变量

- Definition: A literal is either a variable or the negation of a variable.
- Example: $\varphi = \neg(a \vee \neg b)$

  Variables: $AP(\varphi) = \{a, b\}$

  Literals: $lit(\varphi) = \{a, \neg b\}$
- Note: Equivalent formulae can have different literals.

  Example: $\varphi' = \neg a \wedge b$

  Literals: $lit(\varphi') = \{\neg a, b\}$

Terms and clauses

- Definition: a term is a conjunction of literals
  - Example: $(a \land \neg b \land c)$
- Definition: a clause is a disjunction of literals
  - Example: $(a \lor \neg b \lor c)$

没有"$\lor$"号，(c)，也算一个 clause

Negation Normal Form (NNF)

A formula is in Negation Normal Form (NNF) iff

1. it contains only $\neg$, $\wedge$ and $\vee$ as connectives and
2. only variables are negated.

Examples:

- $a \rightarrow b$ is not in NNF
- $\neg(a \vee \neg b)$ is not in NNF
- $\neg a \wedge b$ is in NNF

Converting to NNF

- Every formula can be converted to NNF in linear time:
  - eliminate all connectives other than $\wedge$, $\vee$, $\neg$ and
  - use De Morgan and double-negation rules to push negations to operands.

- Example: $\varphi = \neg(a \rightarrow \neg b)$
  - eliminate $'\rightarrow'$: $\varphi = \neg(\neg a \vee \neg b)$
  - push negation using De Morgan: $\varphi = (\neg\neg a \wedge \neg\neg b)$
  - use double-negation rule: $\varphi = (a \wedge b)$
- Tranformation to NNF can be done with an effort (time and space) that is linear in the size of the formula, if the formula does not contain nested if-and-only-if operators.

  Idea: Number of transformation steps $\leq$ number of operands in the formula.

Disjunctive Normal Form (DNF)

- Definition: A formula is said to be in Disjunctive Normal Form (DNF) iff it is a disjunction of terms.
- In other words, it is a formula of the form

$$\bigvee_i \left(\bigwedge_j l_{i,j} \right)$$

where $l_{i,j}$ is the j-th literal in the i-th term.

- Example:

$$\varphi = (a \wedge \neg b \wedge c) \vee (\neg a \wedge d) \vee (b) \quad \textit{is in DNF}$$

- DNF is a special case of NNF.

Converting to DNF

- Every formula can be converted to DNF in exponential time and space:
  1. Convert to NNF
  2. Distribute disjunctions following the rule:
     $\models \varphi_1 \land (\varphi_2 \lor \varphi_3) \leftrightarrow (\varphi_1 \land \varphi_2) \lor (\varphi_1 \land \varphi_3)$
- Example:

$$\begin{aligned}\varphi &= (a \lor b) \land (\neg c \lor d)\\ &= ((a \lor b) \land (\neg c)) \lor ((a \lor b) \land d)\\ &= (a \land \neg c) \lor (b \land \neg c) \lor (a \land d) \lor (b \land d)\end{aligned}$$

- Now consider $\varphi_n = (a_1 \lor b_1) \land (a_2 \lor b_2) \land \ldots \land (a_n \lor b_n)$.
- Q: How many terms will the DNF have?

  A: 2^n

Satisfiability of DNF

- Q: Is the following DNF formula satisfiable?

  $$(a_1 \wedge a_2 \wedge \neg a_1) \vee (a_2 \wedge a_1) \vee (a_2 \wedge \neg a_3 \wedge a_3)$$

  A: Yes, because the term $a_2 \wedge a_1$ is satisfiable.
- Q: What is the complexity of the satisfiability check of DNF formulae?

  A: Linear (time and space).
- Q: Can there be any polynomial transformation into DNF?
- A: No, it would violate the NP-completeness of the problem.

Conjunctive Normal Form (CNF)

- Definition: A formula is said to be in Conjunctive Normal Form (CNF) iff it is a conjunction of clauses.
- In other words, it is a formula of the form

$$\bigwedge_i \left(\bigvee_j l_{i,j} \right)$$

where $l_{i,j}$ is the j-th literal in the i-th clause.

- Example:

$$\varphi = (a \vee \neg b \vee c) \wedge (\neg a \vee d) \wedge (b) \quad \textit{is in CNF}$$

- Also CNF is a special case of NNF.

Converting to CNF

- Every formula can be converted to CNF in exponential time and space:
  1. Convert to NNF
  2. Distribute disjunctions following the rule:
     $\models \varphi_1 \vee (\varphi_2 \wedge \varphi_3) \leftrightarrow (\varphi_1 \vee \varphi_2) \wedge (\varphi_1 \vee \varphi_3)$
- Consider the formula $\varphi = (a_1 \wedge b_1) \vee (a_2 \wedge b_2)$.

  Transformation: $(a_1 \vee a_2) \wedge (a_1 \vee b_2) \wedge (b_1 \vee a_2) \wedge (b_1 \vee b_2)$
- Now consider $\varphi_n = (a_1 \wedge b_1) \vee (a_2 \wedge b_2) \vee \ldots \vee (a_n \wedge b_n)$.

  Q: How many clauses does the resulting CNF have?

  A: 2^n

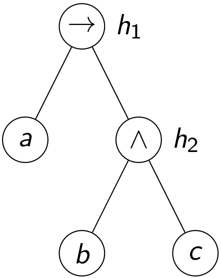
Converting to CNF: Tseitin's encoding

- Every formula can be converted to CNF in linear time and space if new variables are added.
- The original and the converted formulae are not equivalent but equi-satisfiable.

- Consider the formula

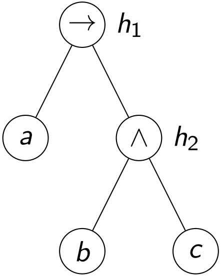
  $\varphi = (a \rightarrow (b \wedge c))$
- Associate a new auxiliary variable with each inner (non-leaf) node.
- Add constraints that define these new variables.
- Finally, enforce the truth of the root node.

Parse tree:



Converting to CNF: Tseitin's encoding

- Tseitin's encoding:
  $(h_1 \leftrightarrow (a \;\to\; h_2))\wedge$
  $(h_2 \leftrightarrow (b \wedge c))\wedge$
  (h_1)



- Each node's encoding has a CNF representation with 3 or 4 clauses.

  $h_1 \leftrightarrow (a \to h_2)$ in CNF: $(h_1 \vee a) \wedge (h_1 \vee \neg h_2) \wedge (\neg h_1 \vee \neg a \vee h_2)$
  $h_2 \leftrightarrow (b \wedge c)$ in CNF: $(\neg h_2 \vee b) \wedge (\neg h_2 \vee c) \wedge (h_2 \vee \neg b \vee \neg c)$

Converting to CNF: Tseitin's encoding

- Let's go back to

  $\varphi_n = (\underbrace{x_1 \wedge y_1}_{h_1}) \vee (\underbrace{x_2 \wedge y_2}_{h_2}) \vee \cdots \vee (\underbrace{x_n \wedge y_n}_{h_n})$ $\stackrel{SAT}{\equiv}$ $h_1 \vee h_2 \cdots \vee h_3$

- With Tseitin's encoding we need:
  - n auxiliary variables $h_1, \ldots, h_n$.
  - Each adds 3 constraints.
  - Top clause: $(h_1 \vee \cdots \vee h_n)$

$\stackrel{SAT}{\equiv} (h_1 \Leftrightarrow (x_1 \wedge y_1))$

$\vee\, (h_2 \Leftrightarrow (x_2 \wedge y_2))$

$\ldots$

$\vee\, (h_n \Leftrightarrow (x_n \wedge y_n))$

$(\neg h_1 \vee x_1) \wedge (\neg h_1 \vee y_1)$

$\wedge\, (\neg$

- Hence, we have
  - $3n + 1$ clauses, instead of 2^n.
  - $3n$ variables rather than $2n$.

Exercise

How many of the following propositional logic formulas are in NNF, DNF resp. CNF?

$\neg(a \vee \neg b)$	-	-	-
$(a \vee \neg b) \wedge (\neg a \vee b)$	NNF	-	CNF
$(x \wedge y) \vee \neg(x \wedge z)$	-	-	-
$a \wedge b \wedge c$	NNF	DNF	CNF
$a \rightarrow b$	-	-	-
$\neg p \vee \neg q$	NNF	DNF	CNF
u	NNF	DNF	CNF

a) NNF:4 DNF:4 CNF:5

b) NNF:4 DNF:3 CNF:4

c) NNF:5 DNF:4 CNF:4

d) NNF:5 DNF:3 CNF:5

Two classes of algorithms for validity

- Q: Is φ satisfiable? (Is $\neg\varphi$ valid?)
- Complexity: NP-Complete (Cook's theorem)
- Two classes of algorithms for finding out:
  - Enumeration of possible solutions (Truth tables etc.)
  - Deduction

The satisfiability problem

- Given a formula φ, is φ satisfiable?

Enumeration the first:

```
Boolean SAT(φ){
    for all α ∈ Assign
        if Eval(α, φ) return true;
    return false;
}
```

Enumeration the second:

Use substitution to eliminate all variables one by one:

$$\exists a.\ \varphi \quad \text{iff} \quad \varphi[0/a] \vee \varphi[1/a]$$

- Q: What is the difference?
  A: Branching on complete vs. partial assignments.

Deduction

A (deductive) proof system consists of a set of axioms and inference rules.

- Inference rules:

$$\frac{\textit{Antecedents}}{\textit{Consequents}} \quad (\textit{rule name})$$

Meaning: If all antecedents hold then at least one of the consequents can be derived.

- Axioms are inference rules with no antecedents, e.g.,

$$\frac{}{a \rightarrow (b \rightarrow a)} \quad (\textit{H1})$$

- Examples:

$$\frac{a \rightarrow b \qquad b \rightarrow c}{a \rightarrow c} \quad (\textit{Trans})$$

$$\frac{a \rightarrow b \qquad a}{b} \quad (\textit{M.P.})$$

Proofs

- Let $\mathcal{H}$ be a proof system.
- $\Gamma \vdash_{\mathcal{H}} \varphi$ means: There is a proof of φ in system $\mathcal{H}$ whose premises are included in Γ
- $\vdash_{\mathcal{H}}$ is called the provability (derivability) relation.

Example

- Let $\mathcal{H}$ be the proof system comprised of the rules Trans and M.P. that we saw earlier:

$$\frac{a \rightarrow b \quad b \rightarrow c}{a \rightarrow c} \qquad (Trans)$$

$$\frac{a \rightarrow b \quad a}{b} \qquad (M.P.)$$

- Does the following relation hold?

$$a \rightarrow b,\ b \rightarrow c,\ c \rightarrow d,\ d \rightarrow e,\ a \quad \vdash_{\mathcal{H}} \quad e$$

Deductive proof: Example

$$\frac{a \to b \quad b \to c}{a \to c} \quad (Trans) \qquad \frac{a \to b \quad a}{b} \quad (M.P.)$$

$$a \to b,\ b \to c,\ c \to d,\ d \to e,\ a \quad \vdash_{\mathcal{H}} \quad e$$

1.	*premise* :	$a \to b$
2.	*premise*	$b \to c$
3.	1, 2, *Trans* :	$a \to c$
4.	*premise* :	$c \to d$
5.	*premise* :	$d \to e$
6.	4, 5, *Trans* :	$c \to e$
7.	3, 6, *Trans* :	$a \to e$
8.	*premise* :	a
9.	7, 8, *M.P.* :	e

$$
\begin{tikzcd}[column sep=small]
a \to b \arrow[dr] & & b \to c \arrow[dl] & & c \to d \arrow[dr] & & d \to e \arrow[dl] & \\
& \overset{\text{trans}}{a \to c} \arrow[drr] & & & & \overset{\text{trans}}{c \to e} \arrow[dll] & & \\
& & & \overset{\text{trans}}{a \to e} \arrow[drr] & & & & a \arrow[dll] \\
& & & & & \overset{\text{M.P.}}{e} & &
\end{tikzcd}
$$

- For a given proof system $\mathcal{H}$,
  - Soundness: Does $\vdash$ conclude "correct" conclusions from premises?
  - Completeness: Can we conclude all true statements with $\mathcal{H}$?
- Correct with respect to what?

  With respect to the semantic definition of the logic. In the case of propositional logic truth tables give us this.

syntactic → semantic

Soundness of $\mathcal{H}$:	*if*	$\Gamma \vdash_{\mathcal{H}} \varphi$	*then*	$\Gamma \models \varphi$
Completeness of $\mathcal{H}$:	*if*	$\Gamma \models \varphi$	*then*	$\Gamma \vdash_{\mathcal{H}} \varphi$

Example: Hilbert axiom system (H)

- Let H be (M.P.) together with the following axiom schemes:

$$\frac{}{a \rightarrow (b \rightarrow a)} \quad (H1)$$

$$\frac{}{((a \rightarrow (b \rightarrow c)) \rightarrow ((a \rightarrow b) \rightarrow (a \rightarrow c)))} \quad (H2)$$

$$\frac{}{(\neg b \rightarrow \neg a) \rightarrow (a \rightarrow b)} \quad (H3)$$

- H is sound and complete for propositional logic.

Proof of soundness and completeness

- To prove soundness of H, prove the soundness of its axioms and inference rules (easy with truth-tables).
  For example:

a	b	$a \rightarrow (b \rightarrow a)$
0	0	1
0	1	1
1	0	1
1	1	1

- Completeness: harder, but possible.

The resolution proof system

- The resolution inference rule for CNF: 一次只能消去一个

Positive negative

$$\frac{(\cancel{l} \vee l_1 \vee l_2 \vee \ldots \vee l_n) \quad (\cancel{\neg l} \vee l'_1 \vee \ldots \vee l'_m)}{(l_1 \vee \ldots \vee l_n \vee l'_1 \vee \ldots \vee l'_m)} \; \textit{Resolution}$$

- Example:

$$\frac{(a \vee b) \quad (\neg a \vee c)}{(b \vee c)}$$

- We first see some example proofs, before proving soundness and completeness.

Proof by resolution

- Let $\varphi = (a_1 \vee a_3) \wedge (\neg a_1 \vee a_2 \vee a_5) \wedge (\neg a_1 \vee a_4) \wedge (\neg a_1 \vee \neg a_4)$
- We want to prove $\varphi \ \rightarrow \ (a_3)$

$$
\begin{tikzcd}
 & & (\neg a_1 \vee \cancel{a_4}) \arrow[dr] & & (\neg a_1 \vee \neg \cancel{a_4}) \arrow[dl] \\
(\cancel{a_1} \vee a_3) \arrow[dr] & & & (\cancel{\neg} a_1) \arrow[dll] & \\
 & (a_3) & & &
\end{tikzcd}
$$

Resolution

- Resolution is a sound and complete proof system for CNF.
- If the input formula is unsatisfiable, there exists a proof of the empty clause.

Let $\varphi = (a_1 \vee a_3) \wedge (\neg a_1 \vee a_2) \wedge (\neg a_1 \vee a_4) \wedge (\neg a_1 \vee \neg a_4) \wedge (\neg a_3)$.

$$
\begin{tikzcd}
(\neg a_1 \vee a_4) \arrow[dr] & & (\neg a_1 \vee \neg a_4) \arrow[dl] & & \\
& (\neg a_1) \arrow[dr] & & (a_1 \vee a_3) \arrow[dl] & \\
& & a_3 \arrow[dr] & & \neg a_3 \arrow[dl] \\
& & & () &
\end{tikzcd}
$$

Soundness and completeness of resolution

- Soundness is straightforward. Just prove by truth table that

$$\models ((\varphi_1 \vee a) \wedge (\varphi_2 \vee \neg a)) \rightarrow (\varphi_1 \vee \varphi_2).$$

- Completeness is a bit more involved.
  Basic idea: Use resolution for variable elimination.

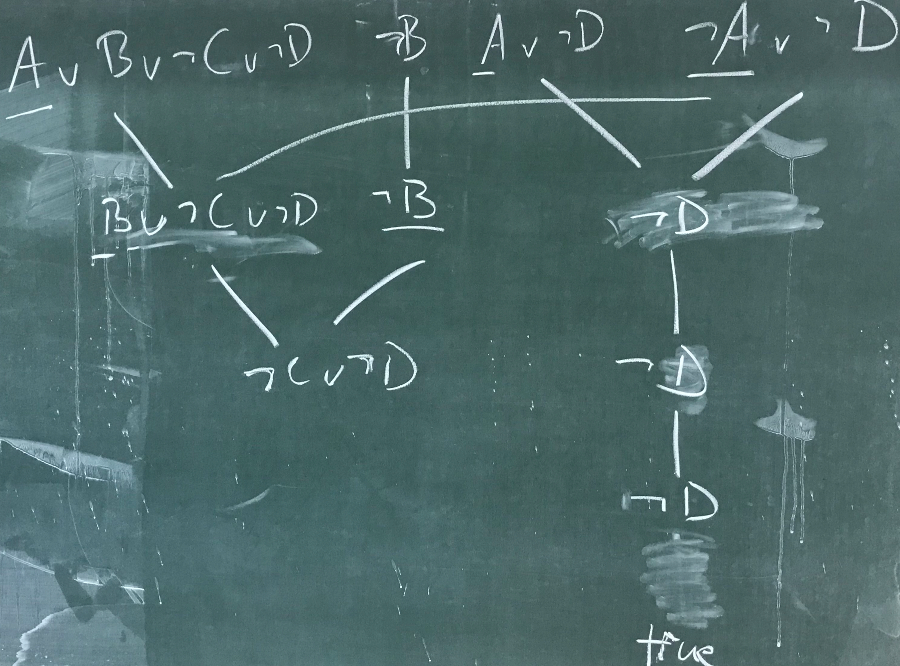
$$\begin{array}{c} (a \vee \varphi_1) \wedge \ldots \wedge (a \vee \varphi_n) \wedge \\ (\neg a \vee \psi_1) \wedge \ldots (\neg a \vee \psi_m) \wedge \\ R \\ \Leftrightarrow \\ (\varphi_1 \vee \psi_1) \wedge \ldots \wedge (\varphi_1 \vee \psi_m) \wedge \\ \ldots \\ (\varphi_n \vee \psi_1) \wedge \ldots (\varphi_n \vee \psi_m) \wedge \\ R \end{array}$$

  where φ_i ($i = 1, \ldots, n$), ψ_j ($j = 1, \ldots, m$), and R contains neither a nor $\neg a$.

Learning target

- How is negation normal form (NNF) defined and how can we convert logical formulas to NNF?
- How is disjunctive normal form (DNF) defined and how can we convert logical formulas to DNF?
- How is conjunctive normal form (CNF) defined and how can we convert logical formulas to CNF?
- What are the advantages and disadvantages of DNF and CNF?

- How can enumeration be used to solve the satisfiability problem?
- What is a deductive proof system and how can we derive proofs in it?
- When is a deductive proof system sound? When is it complete?
- What is resolution?


A ∨ B ∨ ¬C ∨ ¬D
¬B
A ∨ ¬D
¬A ∨ ¬D
B ∨ ¬C ∨ ¬D
¬B
¬C ∨ ¬D
¬D
¬D
¬D
true