Two classes of algorithms for validity

- Q: Is ϕ satisfiable (/$\neg\phi$ is valid) ?
- Complexity: NP-Complete (the first-ever! – Cook's theorem)

- Two classes of algorithms for finding out:
  1. Enumeration of possible solutions (Truth tables etc).
  2. Deduction

- More generally (beyond propositional logic):
  - Enumeration is possible only in some logics.
  - Deduction cannot necessarily be fully automated.

The satisfiability problem: enumeration

- Given a formula ϕ, is ϕ satisfiable?

```
Boolean SAT(ϕ) {
  B:=false
  for all α ∈ 2^AP(ϕ)
    B = B ∨ Eval(ϕ,α)
  end
  return B
}
```

- There must be a better way to do that in practice.

A Brief Introduction to Logic - Outline

- Brief historical notes on logic
- Propositional Logic :Syntax
- Propositional Logic :Semantics
- Satisfiability and validity
- Modeling with Propositional logic
- Normal forms
- Deductive proofs and resolution

Definitions…

- Definition: A literal is either an atom or a negation of an atom.
- Let $\phi = \neg(A \vee \neg B)$. Then:
  - Atoms: $AP(\phi) = \{A,B\}$
  - Literals: $lit(\phi) = \{A, \neg B\}$
- Equivalent formulas can have different literals
  - $\phi = \neg(A \vee \neg B) = \neg A \wedge B$
  - Now $lit(\phi) = \{\neg A, B\}$

Definitions…

- Definition: a term is a conjunction of literals
  - Example: $(A \wedge \neg B \wedge C)$
- Definition: a clause is a disjunction of literals
  - Example: $(A \vee \neg B \vee C)$

Negation Normal Form (NNF)

- Definition: A formula is said to be in Negation Normal Form (NNF) if it only contains $\neg$, $\wedge$ and $\vee$ connectives and only atoms can be negated.

- Examples:
  - $\phi_1 = \neg(A \vee \neg B)$ is not in NNF
  - $\phi_2 = \neg A \wedge B$ is in NNF

Converting to NNF

- Every formula can be converted to NNF in linear time:
  - Eliminate all connectives other than $\wedge$, $\vee$, $\neg$
  - Use De Morgan and double-negation rules to push negations to the right
- Example: $\phi = \neg(A \rightarrow \neg B)$
  - Eliminate '$\rightarrow$': $\phi = \neg(\neg A \vee \neg B)$
  - Push negation using De Morgan: $\phi = (\neg\neg A \wedge \neg\neg B)$
  - Use Double negation rule: $\phi = (A \wedge B)$

Disjunctive Normal Form (DNF)

- Definition: A formula is said to be in Disjunctive Normal Form (DNF) if 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.

- Examples
  - $\phi = (A \wedge \neg B \wedge C) \vee (\neg A \wedge D) \vee (B)$ 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:
  - Convert to NNF
  - Distribute disjunctions following the rule:
    $\models A \wedge (B \vee C) \leftrightarrow ((A \wedge B) \vee (A \wedge C))$
- Example:
  - $\phi = (A \vee B) \wedge (\neg C \vee D) =$
    $((A \vee B) \wedge (\neg C)) \vee ((A \vee B) \wedge D) =$
    $(A \wedge \neg C) \vee (B \wedge \neg C) \vee (A \wedge D) \vee (B \wedge D)$
  - Q: how many clauses would the DNF have had we started from a conjunction of n clauses ?

Satisfiability of DNF

- Is the following DNF formula satisfiable?

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

- What is the complexity of satisfiability of DNF formulas?

Conjunctive Normal Form (CNF)

- Definition: A formula is said to be in Conjunctive Normal Form (CNF) if 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 term.

- Examples
  - $\phi = (A \vee \neg B \vee C) \wedge (\neg A \vee D) \wedge (B)$ is in CNF
- CNF is a special case of NNF

Converting to CNF

- Every formula can be converted to CNF:
  - in exponential time and space with the same set of atoms
  - in linear time and space if new variables are added.
    - In this case the original and converted formulas are "equi-satisfiable".
    - This technique is called Tseitin's encoding.

Converting to CNF: the exponential way

CNF(ϕ) {

case

ϕ is a literal: return ϕ

ϕ is $\psi_1 \wedge \psi_2$: return CNF(ψ_1) $\wedge$ CNF(ψ_2)

ϕ is $\psi_1 \vee \psi_2$: return Dist(CNF(ψ_1),CNF(ψ_2))

}

Dist(ψ_1,ψ_2) {

case

ψ_1 is $\phi_{11} \wedge \phi_{12}$: return Dist($\phi_{11}$,$\psi_2$) $\wedge$ Dist(ψ_{12},ψ_2)

ψ_2 is $\phi_{21} \wedge \phi_{22}$: return Dist($\psi_1$,$\phi_{21}$) $\wedge$ Dist(ψ_1,ϕ_{22})

else: return $\psi_1 \vee \psi_2$

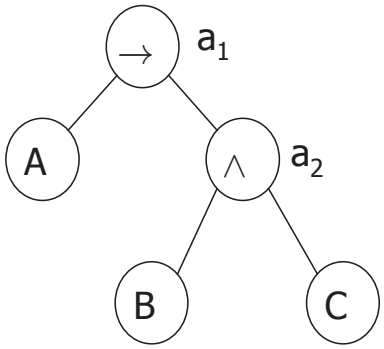
Converting to CNF: the exponential way

- Consider the formula
  $\phi = (x_1 \wedge y_1) \vee (x_2 \wedge y_2)$
- CNF(ϕ)=
  $(x_1 \vee x_2) \wedge$
  $(x_1 \vee y_2) \wedge$
  $(y_1 \vee x_2) \wedge$
  $(y_1 \vee y_2)$

- Now consider: $\phi_n = (x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \cdots \vee (x_n \wedge y_n)$
- Q: How many clauses CNF(ϕ) returns ?
- A: 2^n

Converting to CNF: Tseitin's encoding

- Consider the formula $\phi = (A \rightarrow (B \wedge C))$
- The parse tree:



- Associate a new auxiliary variable with each gate.
- Add constraints that define these new variables.
- Finally, enforce the root node.

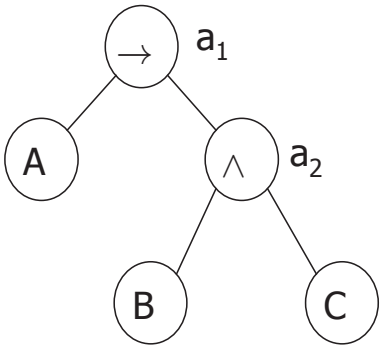
Converting to CNF: Tseitin's encoding

- Need to satisfy:

  $(a_1 \leftrightarrow (A \rightarrow a_2)) \wedge$

  $(a_2 \leftrightarrow (B \wedge C)) \wedge$

  (a_1)



- Each such constraint has a CNF representation with 3 or 4 clauses.

Converting to CNF: Tseitin's encoding

- Need to satisfy:

  $(a_1 \leftrightarrow (A \rightarrow a_2)) \wedge$

  $(a_2 \leftrightarrow (B \wedge C)) \wedge$

  (a_1)

- First: $(a_1 \vee A) \wedge (a_1 \vee \neg a_2) \wedge (\neg a_1 \vee \neg A \vee a_2)$
- Second: $(\neg a_2 \vee B) \wedge (\neg a_2 \vee C) \wedge (a_2 \vee \neg B \vee \neg C)$

Converting to CNF: Tseitin's encoding

- Let's go back to

  $\phi_n = (x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \cdots \vee (x_n \wedge y_n)$

- With Tseitin's encoding we need:
  - n auxiliary variables $a_1,\ldots,a_n$.
  - Each adds 3 constraints.
  - Top clause: $(a_1 \vee \cdots \vee a_n)$

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

What now?

- Time to solve the decision problem for propositional logic.
  - The only algorithm we saw so far was building truth tables.

Two classes of algorithms for validity

- Q: Is ϕ valid ?
  - Equivalently: is $\neg\phi$ satisfiable?
- Two classes of algorithm for finding out:
  1. Enumeration of possible solutions (Truth tables etc).
  2. Deduction

- In general (beyond propositional logic):
  - Enumeration is possible only in some theories.
  - Deduction typically cannot be fully automated.

The satisfiability Problem: enumeration

- Given a formula ϕ, is ϕ satisfiable?

```
Boolean SAT(ϕ) {
   B:=false
   for all α ∈ 2^AP(ϕ)
      B = B ∨ Eval(ϕ,α)
   end
   return B
}
```

- NP-Complete (the first-ever! – Cook's theorem)

A Brief Introduction to Logic - Outline

- Brief historical notes on logic
- Propositional Logic :Syntax
- Propositional Logic :Semantics
- Satisfiability and validity
- Modeling with Propositional logic
- Normal forms
- Deductive proofs and resolution

Deduction requires axioms and Inference rules

- Inference rules:

$$\frac{\text{Antecedents}}{\text{Consequent}} \quad \text{(rule-name)}$$

- Examples:

$$\frac{A \rightarrow B \quad B \rightarrow C}{A \rightarrow C} \quad \text{(trans)}$$

$$\frac{A \rightarrow B \quad A}{B} \quad \text{(M.P.)}$$

Axioms

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

$$\frac{}{A \rightarrow (B \rightarrow A)}\ (H1)$$

- We can turn an inference rule into an axiom if we have '$\rightarrow$' in the logic.

- So the difference between them is not sharp.

Proofs

- A proof uses a given set of inference rules and axioms.
- This is called the *proof system*.
- Let $\mathcal{H}$ be a proof system.

- $\Gamma \vdash_{\mathcal{H}} \phi$ means: there is a proof of ϕ in system $\mathcal{H}$ whose premises are included in Γ

- $\vdash_{\mathcal{H}}$ is called the provability relation.

Example

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

- Does the following relation holds?

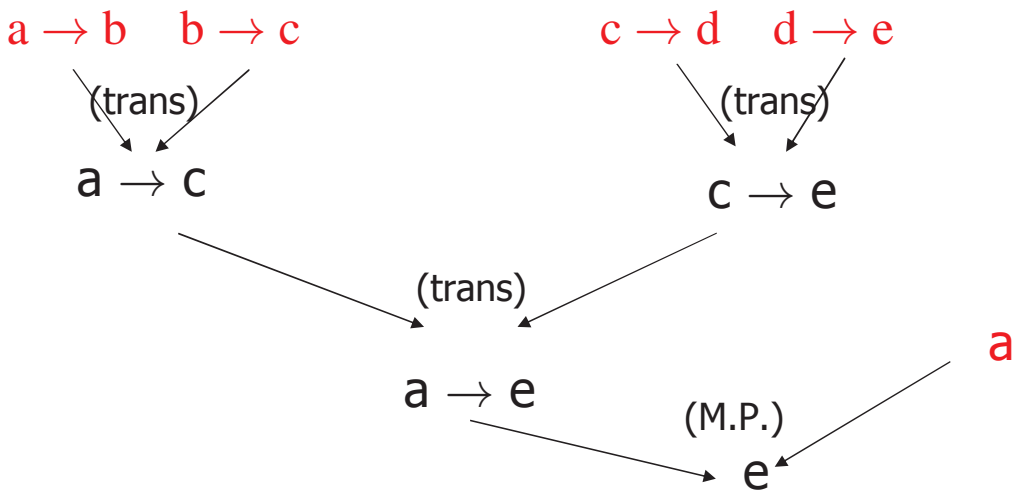
$$a \to b, b \to c,\ c \to d,\ d \to e,\ a \vdash_{\mathcal{H}} e$$

Deductive proof: example

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

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

Proof graph (DAG)



Roots: premises

Proofs

- The problem: $\vdash$ is a relation defined by syntactic transformations of the underlying proof system.
- For a given proof system $\mathcal{H}$,
  - does $\vdash$ conclude "correct" conclusions from premises ?
  - 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 gives us this.

Soundness and completeness

- Let $\mathcal{H}$ be a proof system
- *Soundness* of $\mathcal{H}$: if $\vdash_{\mathcal{H}} \phi$ then $\models \phi$
- *Completeness* of $\mathcal{H}$: if $\models \phi$ then $\vdash_{\mathcal{H}} \phi$
- How to prove soundness and completeness ?

Example: Hilbert axiom system ($\mathcal{H}$)

- Let $\mathcal{H}$ be (M.P) + the following axiom schemas:

$$\frac{}{A \rightarrow (B \rightarrow A)} \quad \text{(H1)}$$

$$\frac{}{((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))} \quad \text{(H2)}$$

$$\frac{}{(\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B)} \quad \text{(H3)}$$

- $\mathcal{H}$ is sound and complete

Soundness and completeness

- To prove soundness of $\mathcal{H}$, 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 inference system

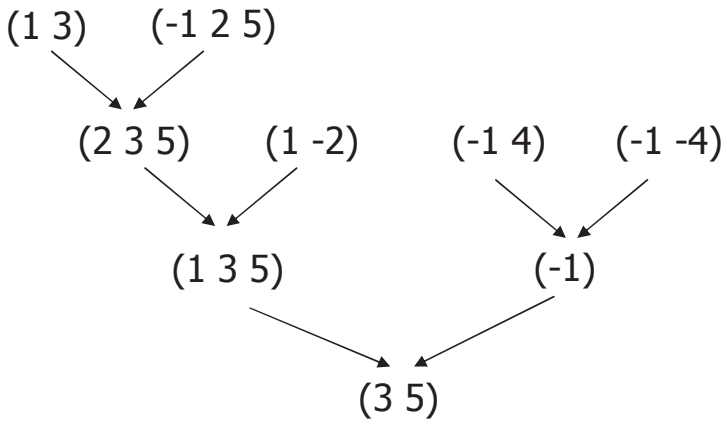
- The resolution inference rule for CNF:

$$\frac{(l \vee l_1 \vee .. \vee l_n) \qquad (\neg l \vee l'_1 \vee .. \vee l'_n)}{(l_1 \vee .. \vee l_n \vee l'_1 \vee .. \vee l'_n)} \text{ (Resolution)}$$

- Example: $\dfrac{(a \vee b) \qquad (\neg a \vee c)}{(b \vee c)}$

Proof by resolution

- Let $\varphi = (1\ 3) \wedge (-1\ 2\ 5) \wedge (-1\ 4) \wedge (-1\ -4)$
- We'll try to prove $\varphi \rightarrow (3\ 5)$

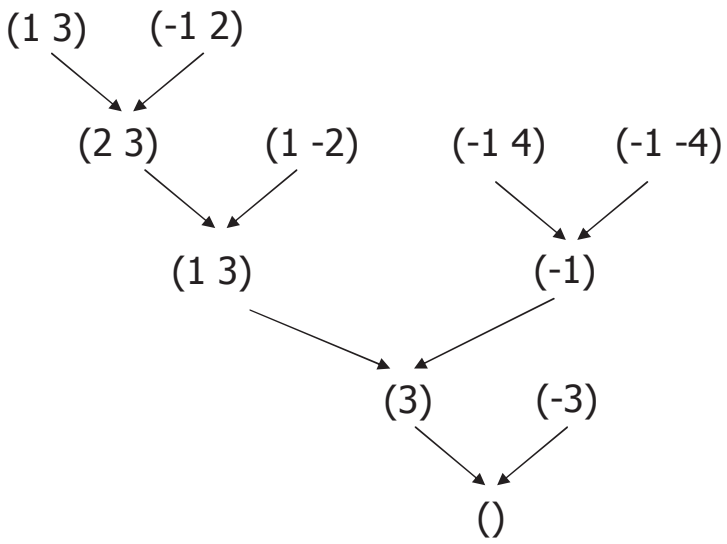


Resolution

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

Example

- Let $\varphi = (1\ 3) \wedge (-1\ 2) \wedge (-1\ 4) \wedge (-1\ -4) \wedge (-3)$



Another system: Natural deduction ($\mathcal{N}$)

- $\dfrac{A \quad B}{A \wedge B}$ (Introduction-and) (I - $\wedge$)

- $\dfrac{A \wedge B}{A}$ (Elimination-1-and) (E1 - $\wedge$)

- $\dfrac{A \wedge B}{B}$ (Elimination-2-and) (E2 - $\wedge$)

Example

- Theorem: $p \wedge q, r \vdash_{\mathcal{N}} q \wedge r$
- Note: the theorem only claims provability relation. Correctness is implied by the soundness of $\mathcal{N}$.
- Proof:

1.	$p \wedge q$	premise
2.	r	premise
3.	q	E-2-$\wedge$, 1
4.	$q \wedge r$	I-and, 2,3

More rules for $\mathcal{N}$

- $\dfrac{\neg\neg A}{A}$ (E-double negation)
- $\dfrac{A}{\neg\neg A}$ (I-double negation)

More rules for $\mathcal{N}$

- $\dfrac{A \quad A \rightarrow B}{B}$ (E - implication)
- Similar to another elimination rule:
- $\dfrac{\neg B \quad A \rightarrow B}{\neg A}$ (Modus-Tollens (M.T.))
- If assuming p allows to prove q then
  $p \rightarrow q$ (I – implication)

Example

- Theorem: $p \rightarrow q \vdash \neg q \rightarrow \neg p$
- Proof:

1.	$p \wedge q$	premise
2.	$\neg q$	assumption
3.	$\neg p$	M.T. 1,2
4.	$\neg q \rightarrow \neg p$	I-implication

- Note the difference between assumptions and premises. The former needs to be discharged.
- The introduction-implication rule lets us discharge assumptions.

Example

- Theorem: $\neg q \rightarrow \neg p \vdash p \rightarrow \neg\neg q$
- Proof:

1.	$\neg q \rightarrow \neg p$	premise
2.	p	assumption
3.	$\neg\neg q$	M.T. 1,2
4.	$p \rightarrow \neg\neg q$	I-implication

Example

- Theorem: $\vdash (q \rightarrow r) \rightarrow ((\neg q \rightarrow \neg p) \rightarrow (p \rightarrow r))$
- Proof:

1.	$q \rightarrow r$	assumption
2.	$\neg q \rightarrow \neg p$	assumption
3.	p	assumption
4.	$\neg\neg p$	I-double-negation 3
5.	$\neg\neg q$	M.T., 2,4
6.	q	E-double-negation,5
7.	r	E-implication 1,6
8.	$p \rightarrow r$	I-implication 3-7
9.	$(\neg q \rightarrow \neg p) \rightarrow (p \rightarrow r)$	I-implication 2-8
10.	$(q \rightarrow r) \rightarrow ((\neg q \rightarrow \neg p) \rightarrow (p \rightarrow r)$	I-implication 1-9

More rules...

- $\dfrac{A \quad .}{A \vee B}$ (I-or1)
- $\dfrac{B \quad .}{A \vee B}$ (I-or2)

- $\dfrac{(p \vee q)\ (p \rightarrow r)\ (q \rightarrow r)}{r}$ (E-or)

Example

- Theorem: $p \vee q \vdash q \vee p$
- Proof:

1.	$p \vee q$	premise
2.	p	assumption
3.	$q \vee p$	I-or1
4.	$p \rightarrow q \vee p$	I-implication
5.	q	assumption
6.	$q \vee p$	I-or2
7.	$q \rightarrow q \vee p$	I-implication
8.	$q \vee p$	E-or