# Propositional Logic

Bow-Yaw Wang

Institute of Information Science
Academia Sinica, Taiwan

September 12, 2017

# Outline

- Consider the following arguments:

**Example**

若火車誤點且車站沒有計程車，則小明開會就遲到。小明開會並沒有遲到，而火車誤點。那麼車站就有計程車。

**Example**

如果下雨而且小華沒帶雨傘，則小華會淋溼。小華並沒有淋溼，而外面正在下雨。那麼小華一定帶了雨傘。

- Both examples have the same structure:

| | | |
|---|---|---|
| $p$ | 火車誤點 | 下雨 |
| $q$ | 車站有計程車 | 小華帶雨傘 |
| $r$ | 小明開會遲到 | 小華淋溼 |

If $p$ and not $q$, then $r$. Not $r$. $p$. Hence $q$.
（若$p$ 且非$q$，則$r$。非$r$，$p$。則$q$）

# Propositions

- We will develop a language to reason such arguments.
- Our langauge is based on propositions (or declarative sentences).
- Examples:
  - The sum of 3 and 5 equals 8.
  - Every even natural number is the sum of two prime numbers (Goldbach's conjecture).
  - All hobbits like mushrooms in their soup.
- A proposition can either be "true" or "false."
- Non-examples:
  - When will we have lunch?
  - Run!

# Atomic Sentences

- Certain sentences are the basic blocks of our language.
  - They are called atomic (or indecomposable) sentences.
- We will use $p, q, r, \ldots$ (possibly with sub- or super-scripts) to denote sentences.
- Examples:
  - Let $p$ denote "I won the lottery last week."
  - Let $q$ denote "I bought a lottery ticket."
  - Let $r$ denote "I won last week's grand prize."
- In fact, $p$, $q$, and $r$ are all atomic sentences.

# Sentences

- Let $p, q, r, \ldots$ be sentences.
  - $p$ : "I won the lottery last week."
  - $q$ : "I bought a lottery ticket."
  - $r$ : "I won last week's grand prize."
- We construct new sentences by the following connectives:
  - The negation of $p$ (denoted by $\neg p$).
    - ★ It is **not** true that "I won the lottery last week."
  - The disjunction of $p$ and $q$ (denoted by $p \vee q$).
    - ★ "I won the lottery last week" **or** "I won last week's grand prize."
  - The conjunction of $p$ and $q$ (denoted by $p \wedge q$).
    - ★ "I won the lottery last week" **and** "I bought a lottery ticket."
  - The implication of $r$ and $p$ (denoted by $r \Longrightarrow p$).
    - ★ "I won last week's grand prize" **implies** "I won the lottery last week."

# Binding Priorities

- If $p, q, r$ are sentences, $p \wedge q$ and $(\neg r) \vee q$ are sentences.
- $(p \wedge q) \Longrightarrow ((\neg r) \vee q)$ is also a sentence.
- To reduce the number of parentheses, we adopt the following conventions:
  **Convention**.

| strong | | weak |
|---|---|---|
| $\neg$ | $\{\vee, \wedge\}$ | $\Longrightarrow$ |

- Hence $p \wedge q \Longrightarrow \neg r \vee q$ is indeed $(p \wedge q) \Longrightarrow ((\neg r) \vee q)$.

# Examples, Examples, Examples

- Let us rewrite our examples:

**Example**

若火車誤點且車站沒有計程車，則小明開會就遲到。小明開會並沒有遲到，而火車誤點。那麼車站就有計程車。

- We have the following atomic sentences:
  $p$： 火車誤點 | $q$： 車站有計程車 | $r$： 小明開會遲到
- In our language, we write:
  - $p \wedge \neg q \Longrightarrow r$ （若火車誤點且車站沒有計程車，則小明開會就遲到）
  - $\neg r$ （小明開會並沒有遲到）
  - $p$ （火車誤點）
  - Hence $q$ （車站就有計程車）

# Examples, Examples, Examples

- Let us rewrite our examples:

**Example**

如果下雨而且小華沒帶雨傘，則小華會淋溼。小華並沒有淋溼，而外面正在下雨。那麼小華一定帶了雨傘。

- We have the following atomic sentences:
  $p$： 下雨 | $q$： 小華帶雨傘 | $r$： 小華淋溼
- In our language, we write:
  - $p \wedge \neg q \implies r$ （如果下雨而且小華沒帶雨傘，則小華會淋溼）
  - $\neg r$ （小華並沒有淋溼）
  - $p$ （外面正在下雨）
  - Hence $q$ （小華一定帶了雨傘）

# Outline

# Natural Deduction

- In our examples, we (informally) infer new sentences.
- In natural deduction, we have a collection of proof rules.
  - These proof rules allow us to infer new sentences logically followed from existing ones.
- Supose we have a set of sentences: $\phi_1, \phi_2, \ldots, \phi_n$ (called premises), and another sentence $\psi$ (called a conclusion).
- The notation

$$\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$$

  is called a sequent.
- A sequent is valid if a proof (built by the proof rules) can be found.
- We will try to build a proof for our examples. Namely,

$$p \wedge \neg q \implies r, \neg r, p \vdash q.$$

# Proof Rules for Natural Deduction – Conjunction

- Suppose we want to prove a conclusion $\phi \wedge \psi$. What do we do?
  - Of course, we need to prove both $\phi$ and $\psi$ so that we can conclude $\phi \wedge \psi$.
- Hence the proof rule for conjunction is

$$\frac{\phi \quad \psi}{\phi \wedge \psi} \wedge i$$

  - Note that premises are shown above the line and the conclusion is below. Also, $\wedge i$ is the name of the proof rule.
  - This proof rule is called "conjunction-introduction" since we introduce a conjunction ($\wedge$) in the conclusion.

# Proof Rules for Natural Deduction – Conjunction

- For each connective, we have introduction proof rule(s) and also elimination proof rule(s).
- Suppose we want to prove a conclusion $\phi$ from the premise $\phi \wedge \psi$. What do we do?
  - We don't do any thing since we know $\phi$ already!
- Here are the elimination proof rules:

$$\frac{\phi \wedge \psi}{\phi}\ \wedge e_1 \qquad\qquad \frac{\phi \wedge \psi}{\psi}\ \wedge e_2$$

- The rule $\wedge e_1$ says: if you have a proof for $\phi \wedge \psi$, then you have a proof for $\phi$ by applying this proof rule.
- Why do we need two rules?
  - Because we want to manipulate syntax only.

# Examples

## Example

Prove $p \wedge q, r \vdash q \wedge r$.

## Proof.

We are looking for a proof of the form:

$$\begin{array}{c} p \wedge q \quad r \\ \vdots \\ q \wedge r \end{array}$$

# Examples

## Example

Prove $p \wedge q, r \vdash q \wedge r$.

## Proof.

We are looking for a proof of the form:

$$\dfrac{\dfrac{p \wedge q}{q}\ \wedge e_2 \qquad r}{q \wedge r}\ \wedge i$$

We will write proofs in lines:

| | | |
|---|---|---|
| 1 | $p \wedge q$ | premise |
| 2 | $r$ | premise |
| 3 | $q$ | $\wedge e_2$ 1 |
| 4 | $q \wedge r$ | $\wedge i$ 3, 2 |

# Proof Rules for Natural Deduction – Double Negation

- Suppose we want to prove $\phi$ from a proof for $\neg\neg\phi$. What do we do?
  - There is no difference between $\phi$ and $\neg\neg\phi$. The same proof suffices!
- Hence we have the following proof rules:

$$\frac{\phi}{\neg\neg\phi}\ \neg\neg i \qquad \frac{\neg\neg\phi}{\phi}\ \neg\neg e$$

# Examples

## Example

Prove $p, \neg\neg(q \wedge r) \vdash \neg\neg p \wedge r$.

## Proof.

We are looking for a proof like:

$$\begin{array}{cc} p & \neg\neg(q \wedge r) \\ & \vdots \\ & \neg\neg p \wedge r \end{array}$$

# Examples

## Example

Prove $p, \neg\neg(q \wedge r) \vdash \neg\neg p \wedge r$.

## Proof.

We are looking for a proof like:

$$\dfrac{\dfrac{p}{\neg\neg p}\,\neg\neg i \qquad \dfrac{\dfrac{\neg\neg(q \wedge r)}{q \wedge r}\,\neg\neg e}{r}\,\wedge e_2}{\neg\neg p \wedge r}\,\wedge i$$

# Examples

## Example

Prove $p, \neg\neg(q \wedge r) \vdash \neg\neg p \wedge r$.

## Proof.

We are looking for a proof like:

| | | |
|---|---|---|
| 1 | $p$ | premise |
| 2 | $\neg\neg(q \wedge r)$ | premise |
| 3 | $\neg\neg p$ | $\neg\neg i$ 1 |
| 4 | $q \wedge r$ | $\neg\neg e$ 2 |
| 5 | $r$ | $\wedge e_2$ 4 |
| 6 | $\neg\neg p \wedge r$ | $\wedge i$ 3, 5 |

# Proof Rules for Natural Deduction – Implication

- Suppose we want to prove $\psi$ from proofs for $\phi$ and $\phi \Longrightarrow \psi$. What do we do?
  - We just put the two proofs for $\phi$ and $\phi \Longrightarrow \psi$ together.
- Here is the proof rule:

$$\frac{\phi \quad \phi \Longrightarrow \psi}{\psi} \Longrightarrow e$$

- This proof rule is also called *modus ponens*.
- Here is another proof rule related to implication:

$$\frac{\phi \Longrightarrow \psi \quad \neg\psi}{\neg\phi} MT$$

- This proof rule is called *modus tollens*.

# Example

## Example

Prove $p \Longrightarrow (q \Longrightarrow r), p, \neg r \vdash \neg q$.

## Proof.

| | | |
|---|---|---|
| 1 | $p \Longrightarrow (q \Longrightarrow r)$ | premise |
| 2 | $p$ | premise |
| 3 | $\neg r$ | premise |
| 4 | $q \Longrightarrow r$ | $\Longrightarrow e$ 2, 1 |
| 5 | $\neg q$ | $MT$ 4, 3 |

# Proof Rules for Natural Deduction – Implication

- Suppose we want to prove $\phi \Longrightarrow \psi$. What do we do?
  - We assume $\phi$ to prove $\psi$. If succeed, we conclude $\phi \Longrightarrow \psi$ without any assumption.
  - Note that $\phi$ is added as an assumption and then removed so that $\phi \Longrightarrow \psi$ does not depend on $\phi$.
- We use "box" to simulate this strategy.
- Here is the proof rule:

$$\frac{\boxed{\begin{array}{c}\phi \\ \vdots \\ \psi\end{array}}}{\phi \Longrightarrow \psi} \Longrightarrow i$$

- At any point in a box, you can only use a sentence $\phi$ before that point. Moreover, no box enclosing the occurrence of $\phi$ has been closed.

# Example

## Example

Prove $\neg q \implies \neg p \vdash p \implies \neg\neg q$.

## Proof.

$$\dfrac{\dfrac{\dfrac{\neg q \implies \neg p \quad \dfrac{p}{\neg\neg p}}{\neg\neg q}\, MT}{\neg\neg i}}{p \implies \neg\neg q} \implies i$$

| | | |
|---|---|---|
| 1 | $\neg q \implies \neg p$ | premise |
| 2 | $p$ | assumption |
| 3 | $\neg\neg p$ | $\neg\neg i$ 2 |
| 4 | $\neg\neg q$ | $MT$ 1, 3 |
| 5 | $p \implies \neg\neg q$ | $\implies i$ 2-4 |

# Theorems

## Example

Prove $\vdash p \implies p$.

## Proof.

| | | |
|---|---|---|
| 1 | $p$ | assumption |
| 2 | $p \implies p$ | $\implies i$ 1 - 1 |

□

In the box, we have $\phi \equiv \psi \equiv p$.

## Definition

A sentence $\phi$ such that $\vdash \phi$ is called a theorem.

# Examples

## Example

Prove $p \wedge q \Longrightarrow r \vdash p \Longrightarrow (q \Longrightarrow r)$.

## Proof.

| | | | | | |
|---|---|---|---|---|---|
| 1 | $p \wedge q \Longrightarrow r$ | premise | | | |
| 2 | $p$ | assumption | ⌉ | | ⌉ |
| 3 | $q$ | assumption | ⌉ | | \| |
| 4 | $p \wedge q$ | $\wedge i$ 2, 3 | \| | | \| |
| 5 | $r$ | $\Longrightarrow e$ 4, 1 | ⌋ | | \| |
| 6 | $q \Longrightarrow r$ | $\Longrightarrow i$ 3-5 | | | ⌋ |
| 7 | $p \Longrightarrow (q \Longrightarrow r)$ | $\Longrightarrow i$ 2-6 | | | |

# Proof Rules for Natural Deduction – Disjunction

- Suppose we want to prove $\phi \vee \psi$. What do we do?
  - We can either prove $\phi$ or $\psi$.
- Here are the proof rules:

$$\frac{\phi}{\phi \vee \psi} \vee i_1 \qquad \frac{\psi}{\phi \vee \psi} \vee i_2$$

  - Note the symmetry with $\wedge e_1$ and $\wedge e_2$.

$$\frac{\phi \wedge \psi}{\phi} \wedge e_1 \qquad \frac{\phi \wedge \psi}{\psi} \wedge e_2$$

  - Can we have a corresponding symmetric elimination rule for disjunction? Recall

$$\frac{\phi \quad \psi}{\phi \wedge \psi} \wedge i$$

# Proof Rules for Natural Deduction – Disjunction

- Suppose we want to prove $\chi$ from $\phi \vee \psi$. What do we do?
  - We assume $\phi$ to prove $\chi$ and then assume $\psi$ to prove $\chi$.
  - If both succeed, $\chi$ is proved from $\phi \vee \psi$ without assuming $\phi$ and $\psi$.
- Here is the proof rule:

$$\dfrac{\phi \vee \psi \qquad \boxed{\begin{array}{c}\phi \\ \vdots \\ \chi\end{array}} \qquad \boxed{\begin{array}{c}\psi \\ \vdots \\ \chi\end{array}}}{\chi}\ \vee e$$

- In addition to nested boxes, we may have parallel boxes in our proofs.

# Example

Recall that our syntax does not admit commutativity.

## Example

Prove $p \vee q \vdash q \vee p$.

## Proof.

$$\dfrac{p \vee q \qquad \boxed{\dfrac{\dfrac{p}{q \vee p}\ \vee i_2}{}} \qquad \boxed{\dfrac{\dfrac{q}{q \vee p}\ \vee i_1}{}}}{q \vee p}\ \vee e$$

| | | | |
|---|---|---|---|
| 1 | $p \vee q$ | premise | |
| 2 | $p$ | assumption | ] |
| 3 | $q \vee p$ | $\vee i_2$ 2 | ] |
| 4 | $q$ | assumption | ] |
| 5 | $q \vee p$ | $\vee i_1$ 4 | ] |
| 6 | $q \vee p$ | $\vee e$ 1, 2-3, 4-5 | |

# Example

## Example

Prove $q \Longrightarrow r \vdash p \vee q \Longrightarrow p \vee r$.

## Proof.

| | | | | | |
|---|---|---|---|---|---|
| 1 | $q \Longrightarrow r$ | premise | | | |
| 2 | $p \vee q$ | assumption | | | ⌉ |
| 3 | $p$ | assumption | ⌉ | | \| |
| 4 | $p \vee r$ | $\vee i_1$ 3 | ⌋ | | \| |
| 5 | $q$ | assumption | ⌉ | | \| |
| 6 | $r$ | $\Longrightarrow e$ 5, 1 | \| | | \| |
| 7 | $p \vee r$ | $\vee i_2$ 6 | ⌋ | | \| |
| 8 | $p \vee r$ | $\vee e$ 2, 3-4, 5-7 | | | ⌋ |
| 9 | $p \vee q \Longrightarrow p \vee r$ | $\Longrightarrow i$ 2-8 | | | |

# Example

## Example

Prove $p \wedge (q \vee r) \vdash (p \wedge q) \vee (p \wedge r)$.

## Proof.

| | | | |
|---|---|---|---|
| 1 | $p \wedge (q \vee r)$ | premise | |
| 2 | $p$ | $\wedge e_1$ 1 | |
| 3 | $q \vee r$ | $\wedge e_2$ 1 | |
| 4 | $q$ | assumption | ⌈ |
| 5 | $p \wedge q$ | $\wedge i$ 2, 4 | \| |
| 6 | $(p \wedge q) \vee (p \wedge r)$ | $\vee i_1$ 5 | ⌋ |
| 7 | $r$ | assumption | ⌈ |
| 8 | $p \wedge r$ | $\wedge i$ 2, 7 | \| |
| 9 | $(p \wedge q) \vee (p \wedge r)$ | $\vee i_2$ 8 | ⌋ |
| 10 | $(p \wedge q) \vee (p \wedge r)$ | $\vee e$ 3, 4-6, 7-9 | |

# Example

## Example

Prove $(p \wedge q) \vee (p \wedge r) \vdash p \wedge (q \vee r)$.

## Proof.

| | | | |
|---|---|---|---|
| 1 | $(p \wedge q) \vee (p \wedge r)$ | premise | |
| 2 | $p \wedge q$ | assumption | ⎤ |
| 3 | $p$ | $\wedge e_1$ 2 | ⎥ |
| 4 | $q$ | $\wedge e_2$ 2 | ⎥ |
| 5 | $q \vee r$ | $\vee i_1$ 4 | ⎥ |
| 6 | $p \wedge (q \vee r)$ | $\wedge i$ 3, 5 | ⎦ |
| 7 | $p \wedge r$ | assumption | ⎤ |
| 8 | $p$ | $\wedge e_1$ 7 | ⎥ |
| 9 | $r$ | $\wedge e_2$ 7 | ⎥ |
| 10 | $q \vee r$ | $\vee i_2$ 9 | ⎥ |
| 11 | $p \wedge (q \vee r)$ | $\wedge i$ 8, 10 | ⎦ |
| 12 | $p \wedge (q \vee r)$ | $\vee e$ 1, 2-6, 7-11 | |

# Contradiction

## Definition

Contradictions are sentences of the form $\phi \wedge \neg\phi$ or $\neg\phi \wedge \phi$.

- Examples:
  - $p \wedge \neg p$, $\neg(p \vee q \implies r) \wedge (p \vee q \implies r)$.
- Logically, any sentence can be proved from a contradiction.
  - If $0 = 1$, then $100 \neq 100$.
- Particularly, if $\phi$ and $\psi$ are contradictions, we have $\phi \dashv\vdash \psi$.
  - $\phi \dashv\vdash \psi$ means $\phi \vdash \psi$ and $\psi \vdash \phi$ (called provably equivalent).
- Since all contradictions are equivalent, we will use the symbol $\bot$ (called "bottom") for them.
- We are now ready to discuss proof rules for negation.

# Proof Rules for Natural Deduction – Negation

- Since any sentence can be proved from a contradiction, we have

$$\frac{\bot}{\phi}\ \bot e$$

- When both $\phi$ and $\neg\phi$ are proved, we have a contradiction.

$$\frac{\phi \quad \neg\phi}{\bot}\ \neg e$$

  - The proof rule could be called $\bot i$. We use $\neg e$ because it eliminates a negation.

# Example

## Example

Prove $\neg p \vee q \vdash p \implies q$.

## Proof.

| | | | | |
|---|---|---|---|---|
| 1 | $\neg p \vee q$ | premise | | |
| 2 | $\neg p$ | assumption | | ] |
| 3 | $p$ | assumption | ] | \| |
| 4 | $\bot$ | $\neg e$ 3, 2 | \| | \| |
| 5 | $q$ | $\bot e$ 4 | ] | \| |
| 6 | $p \implies q$ | $\implies i$ 3-5 | | ] |
| 7 | $q$ | assumption | | ] |
| 8 | $p$ | assumption | ] | \| |
| 9 | $q$ | copy 7 | ] | \| |
| 10 | $p \implies q$ | $\implies i$ 8-9 | | ] |
| 11 | $p \implies q$ | $\vee e$ 1, 2-6, 7-10 | | |

# Proof Rules for Natural Deduction – Negation

- Suppose we want to prove $\neg\phi$. What do we do?
  - We assume $\phi$ and try to prove a contradiction. If succeed, we prove $\neg\phi$.
- Here is the proof rule:

$$\dfrac{\boxed{\begin{array}{c}\phi\\ \vdots\\ \bot\end{array}}}{\neg\phi}\ \neg i$$

# Example

## Example

Prove $p \Longrightarrow q, p \Longrightarrow \neg q \vdash \neg p$.

## Proof.

| | | | |
|---|---|---|---|
| 1 | $p \Longrightarrow q$ | premise | |
| 2 | $p \Longrightarrow \neg q$ | premise | |
| 3 | $p$ | assumption | ⌈ |
| 4 | $q$ | $\Longrightarrow e$ 3, 1 | \| |
| 5 | $\neg q$ | $\Longrightarrow e$ 3, 2 | \| |
| 6 | $\bot$ | $\neg e$ 4, 5 | ⌋ |
| 7 | $\neg p$ | $\neg i$ 3-6 | |

# Example

## Example

Prove $p \wedge \neg q \implies r, \neg r, p \vdash q$.

## Proof.

| | | | |
|---|---|---|---|
| 1 | $p \wedge \neg q \implies r$ | premise | |
| 2 | $\neg r$ | premise | |
| 3 | $p$ | premise | |
| 4 | $\neg q$ | assumption | ⌉ |
| 5 | $p \wedge \neg q$ | $\wedge i$ 3, 4 | \| |
| 6 | $r$ | $\implies e$ 5, 1 | \| |
| 7 | $\bot$ | $\neg e$ 6, 2 | ⌋ |
| 8 | $\neg\neg q$ | $\neg i$ 4-7 | |
| 9 | $q$ | $\neg\neg e$ 8 | |

# Derived Rules

- Some rules can actually be derived from others.

## Examples

Prove $p \Longrightarrow q, \neg q \vdash \neg p$ (modus tollens).

## Proof.

| | | | |
|---|---|---|---|
| 1 | $p \Longrightarrow q$ | premise | |
| 2 | $\neg q$ | premise | |
| 3 | $p$ | assumption | ⌈ |
| 4 | $q$ | $\Longrightarrow e$ 3, 1 | \| |
| 5 | $\bot$ | $\neg e$ 4, 2 | ⌋ |
| 6 | $\neg p$ | $\neg i$ 3-5 | |

□

# Derived Rules

## Examples

Prove $p \vdash \neg\neg p$ $(\neg\neg i)$

## Proof.

| | | | |
|---|---|---|---|
| 1 | $p$ | premise | |
| 2 | $\neg p$ | assumption | ] |
| 3 | $\bot$ | $\neg e$ 1, 2 | ] |
| 4 | $\neg\neg p$ | $\neg i$ 2-3 | |

- These rules can be replaced by their proofs and are not necessary.
  - They are just macros to help us write shorter proofs.

# Reductio ad absurdum (RAA)

## Example

Prove $\neg p \implies \bot \vdash p$ (RAA).

## Proof.

| | | | |
|---|---|---|---|
| 1 | $\neg p \implies \bot$ | premise | |
| 2 | $\neg p$ | assumption | ⌉ |
| 3 | $\bot$ | $\implies e$ 2, 1 | ⌋ |
| 4 | $\neg\neg p$ | $\neg i$ 2-3 | |
| 5 | $p$ | $\neg\neg e$ 4 | |

# Tertium non datur, Law of the Excluded Middle (LEM)

## Example

Prove $\vdash p \vee \neg p$.

## Proof.

| | | | | |
|---|---|---|---|---|
| 1 | $\neg(p \vee \neg p)$ | assumption | | ] |
| 2 | $p$ | assumption | ] | \| |
| 3 | $p \vee \neg p$ | $\vee i_1$ 2 | \| | \| |
| 4 | $\bot$ | $\neg e$ 3, 1 | ] | \| |
| 5 | $\neg p$ | $\neg i$ 2-4 | | \| |
| 6 | $p \vee \neg p$ | $\vee i_2$ 5 | | \| |
| 7 | $\bot$ | $\neg e$ 6, 1 | | ] |
| 8 | $\neg\neg(p \vee \neg p)$ | $\neg i$ 1-7 | | |
| 9 | $p \vee \neg p$ | $\neg\neg e$ 8 | | |

# Proof Rules for Natural Deduction (Summary)

Conjunction ($\wedge$)

$$\frac{\phi \quad \psi}{\phi \wedge \psi}\ \wedge i \qquad \frac{\phi \wedge \psi}{\phi}\ \wedge e_1 \qquad \frac{\phi \wedge \psi}{\psi}\ \wedge e_2$$

Disjunction ($\vee$)

$$\frac{\phi}{\phi \vee \psi}\ \vee i_1 \qquad \frac{\psi}{\phi \vee \psi}\ \vee i_2$$

$$\frac{\phi \vee \psi \qquad \boxed{\begin{array}{c}\phi\\ \vdots\\ \chi\end{array}} \qquad \boxed{\begin{array}{c}\psi\\ \vdots\\ \chi\end{array}}}{\chi}\ \vee e$$

Implication ($\Longrightarrow$)

$$\frac{\boxed{\begin{array}{c}\phi\\ \vdots\\ \psi\end{array}}}{\phi \Longrightarrow \psi}\ \Longrightarrow i \qquad \frac{\phi \quad \phi \Longrightarrow \psi}{\psi}\ \Longrightarrow e$$

# Proof Rules for Natural Deduction (Summary)

Negation ($\neg$)

$$\dfrac{\boxed{\begin{array}{c}\phi\\ \vdots\\ \bot\end{array}}}{\neg\phi}\ \neg i \qquad\qquad \dfrac{\phi \quad \neg\phi}{\bot}\ \neg e$$

Contradiction ($\bot$)

(no introduction rule) $\qquad \dfrac{\bot}{\phi}\ \bot e$

Double negation ($\neg\neg$)

(no introduction rule) $\qquad \dfrac{\neg\neg\phi}{\phi}\ \neg\neg e$

# Useful Derived Proof Rules

$$\frac{\phi \Longrightarrow \psi \quad \neg\psi}{\neg\phi}\ MT$$

$$\frac{\phi}{\neg\neg\phi}\ \neg\neg i$$

$$\frac{\begin{array}{|c|}\hline \neg\phi \\ \vdots \\ \bot \\ \hline \end{array}}{\phi}\ RAA$$

$$\frac{}{\phi \vee \neg\phi}\ LEM$$

# Provable Equivalence

- Recall $p \dashv\vdash q$ means $p \vdash q$ and $q \vdash p$.
- Here are some provably equivalent sentences:

$$\begin{array}{rcl}
\neg(p \wedge q) & \dashv\vdash & \neg q \vee \neg p \\
\neg(p \vee q) & \dashv\vdash & \neg q \wedge \neg p \\
p \Longrightarrow q & \dashv\vdash & \neg q \Longrightarrow \neg p \\
p \Longrightarrow q & \dashv\vdash & \neg p \vee q \\
p \wedge q \Longrightarrow p & \dashv\vdash & r \vee \neg r \\
p \wedge q \Longrightarrow r & \dashv\vdash & p \Longrightarrow (q \Longrightarrow r)
\end{array}$$

- Try to prove them.

# Proof by Contradiction

- Although it is very useful, the proof rule RAA is a bit puzzling.

$$\dfrac{\boxed{\begin{array}{c}\neg\phi\\ \vdots\\ \bot\end{array}}}{\phi}\ RAA$$

- Instead of proving $\phi$ directly, the proof rule allows indirect proofs.
  - If $\neg\phi$ leads to a contradiction, then $\phi$ must hold.
- Note that indirect proofs are not "constructive."
  - We do not show why $\phi$ holds; we only know $\neg\phi$ is impossible.
- In early 20th century, some logicians and mathematicians chose not to prove indirectly. They are intuitionistic logicians or mathematicians.
- For the same reason, intuitionists also reject

$$\dfrac{}{\phi \vee \neg\phi}\ LEM \qquad\qquad \dfrac{\neg\neg\phi}{\phi}\ \neg\neg e$$

# Proof by Contradiction

## Theorem

*There are* $a, b \in \mathbb{R} \setminus \mathbb{Q}$ *such that* $a^b \in \mathbb{Q}$.

## Proof.

Let $b = \sqrt{2}$. There are two cases:

- If $b^b \in \mathbb{Q}$, we are done since $\sqrt{2} \in \mathbb{R} \setminus \mathbb{Q}$.
- If $b^b \notin \mathbb{Q}$, choose $a = b^b = \sqrt{2}^{\sqrt{2}}$. Then $a^b = \sqrt{2}^{\sqrt{2} \cdot \sqrt{2}} = \sqrt{2}^2 = 2$. Since $\sqrt{2}^{\sqrt{2}}, \sqrt{2} \in \mathbb{R} \setminus \mathbb{Q}$, we are done.

□

- An intuitionist would criticize the proof since it does not tell us what $a, b$ give $a^b \in \mathbb{Q}$.
  - We know $(a, b)$ is either $(\sqrt{2}, \sqrt{2})$ or $(\sqrt{2}^{\sqrt{2}}, \sqrt{2})$.

# Outline

# Well-Formedness

## Definition

A well-formed formula is constructed by applying the following rules finitely many times:

- atom: Every propositional atom $p, q, r, \ldots$ is a well-formed formula;
- $\neg$: If $\phi$ is a well-formed formula, so is $(\neg\phi)$;
- $\wedge$: If $\phi$ and $\psi$ are well-formed formulae, so is $(\phi \wedge \psi)$;
- $\vee$: If $\phi$ and $\psi$ are well-formed formulae, so is $(\phi \vee \psi)$;
- $\Longrightarrow$ : If $\phi$ and $\psi$ are well-formed formulae, so is $(\phi \Longrightarrow \psi)$.

- More compactly, well-formed formulae are defined by the following grammar in Backus Naur form (BNF):

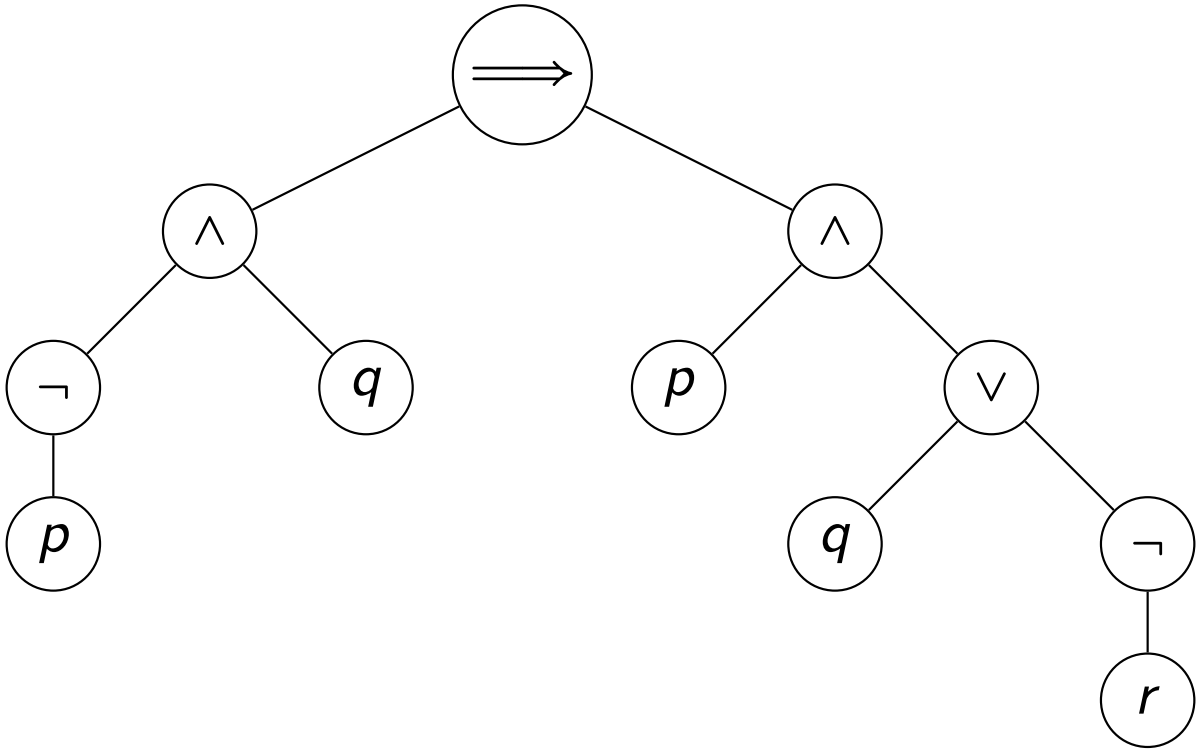
$$\phi ::= p \mid (\neg\phi) \mid (\phi \wedge \phi) \mid (\phi \vee \phi) \mid (\phi \Longrightarrow \phi)$$

# Inversion Principle

- How do we check if $(((\neg p) \wedge q) \implies (p \wedge (q \vee (\neg r))))$ is well-formed?
- Although a well-formed formula needs five grammar rules to construct, the construction process can always be inverted.
  - This is called inversion principle.
- To show $(((\neg p) \wedge q) \implies (p \wedge (q \vee (\neg r))))$ is well-formed, we need to show both $((\neg p) \wedge q)$ and $(p \wedge (q \vee (\neg r)))$ are well-formed.
- To show $((\neg p) \wedge q)$ is well-formed, we need to show both $(\neg p)$ and $q$ are well-formed.
  - $q$ is well-formed since it is an atom.
- To show $(\neg p)$ is well-formed, we need to show $p$ is well-formed.
  - $p$ is well-formed since it is an atom.
- Similarly, we can show $(p \wedge (q \vee (\neg r)))$ is well-formed.

# Parse Tree

- The easiest way to decide whether a formula is well-formed is perhaps by drawing its parse tree.



# Subformulae

- Given a well-formed formula, its subformulae are the well-formed formulae corresponding to its parse tree.
- For instance, the subformulae of the well-formed formulae $(((\neg p) \wedge q) \implies (p \wedge (q \vee (\neg r))))$ are

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

# Outline

# From $\vdash$ to $\models$

- We have developed a calculus to determine whether $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ is valid.
  - That is, from the premises $\phi_1, \phi_2, \ldots, \phi_n$, we can conclude $\psi$.
  - Our calculus is syntactic. It depends on the syntactic structures of $\phi_1, \phi_2, \ldots, \phi_n$, and $\psi$.
- We will introduce another relation between premises $\phi_1, \phi_2, \ldots, \phi_n$ and a conclusion $\psi$.

$$\phi_1, \phi_2, \ldots, \phi_n \models \psi.$$

  - The new relation is defined by 'truth values' of atomic formulae and the semantics of logical connectives.

# Truth Values and Models

## Definition

The set of truth values is $\{\mathsf{F}, \mathsf{T}\}$ where F represents 'false' and T represents 'true.'

## Definition

A valuation or model of a formula $\phi$ is an assignment from each proposition atom in $\phi$ to a truth value.

# Truth Values of Formulae

## Definition

Given a valuation of a formula $\phi$, the truth value of $\phi$ is defined inductively by the following truth tables:

| $\phi$ | $\psi$ | $\phi \wedge \psi$ |
|---|---|---|
| F | F | F |
| F | T | F |
| T | F | F |
| T | T | T |

| $\phi$ | $\psi$ | $\phi \vee \psi$ |
|---|---|---|
| F | F | F |
| F | T | T |
| T | F | T |
| T | T | T |

| $\phi$ | $\psi$ | $\phi \Longrightarrow \psi$ |
|---|---|---|
| F | F | T |
| F | T | T |
| T | F | F |
| T | T | T |

| $\phi$ | $\neg\phi$ |
|---|---|
| F | T |
| T | F |

| $\top$ |
|---|
| T |

| $\bot$ |
|---|
| F |

# Example

- $\phi \wedge \psi$ is T when $\phi$ and $\psi$ are T.
- $\phi \vee \psi$ is F when $\phi$ or $\psi$ is T. wrong
- $\bot$ is always F; $\top$ is always T.
- $\phi \implies \psi$ is T when $\phi$ "implies" $\psi$.

## Example

Consider the valuation $\{q \mapsto \mathsf{T}, p \mapsto \mathsf{F}, r \mapsto \mathsf{F}\}$ of $(q \wedge p) \implies r$. What is the truth value of $(q \wedge p) \implies r$?

## Proof.

Since the truth values of $q$ and $p$ are T and F respectively, the truth value of $q \wedge p$ is F. Moreover, the truth value of $r$ is F. The truth value of $(q \wedge p) \implies r$ is T. □

# Truth Tables for Formulae

- Given a formula $\phi$ with propositional atoms $p_1, p_2, \ldots, p_n$, we can construct a truth table for $\phi$ by listing $2^n$ valuations of $\phi$.

## Example

Find the truth table for $(p \implies \neg q) \implies (q \vee \neg p)$.

## Proof.

| $p$ | $q$ | $\neg p$ | $\neg q$ | $p \implies \neg q$ | $q \vee \neg p$ | $(p \implies \neg q) \implies (q \vee \neg p)$ |
|---|---|---|---|---|---|---|
| F | F | T | T | T | T | T |
| F | T | T | F | T | T | T |
| T | F | F | T | T | F | F |
| T | T | F | F | F | T | T |

# Outline

# Validity of Sequent Revisited

- Informally $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ is valid if we can derive $\psi$ with assumptions $\phi_1, \phi_2, \ldots, \phi_n$.
  - We have formalized "deriving $\psi$ with assumptions $\phi_1, \phi_2, \ldots, \phi_n$" by "constructing a proof in a formal calculus."
- We can give another interpretation by valuations and truth values.
- Consider a valuation $\nu$ over all propositional atoms in $\phi_1, \phi_2, \ldots, \phi_n, \psi$.
  - By "assumptions $\phi_1, \phi_2, \ldots, \phi_n$," we mean "$\phi_1, \phi_2, \ldots, \phi_n$ are T under the valuation $\nu$.
  - By "deriving $\psi$,", we mean $\psi$ is also T under the valuation $\nu$.
- Hence, "we can derive $\psi$ with assumptions $\phi_1, \phi_2, \ldots, \phi_n$" actually means "if $\phi_1, \phi_2, \ldots, \phi_n$ are T under a valuation, then $\psi$ must be T under the same valuation.

# Semantic Entailment

## Definition

We say

$$\phi_1, \phi_2, \ldots, \phi_n \models \psi$$

holds if for every valuations where $\phi_1, \phi_2, \ldots, \phi_n$ are T, $\psi$ is also T. In this case, we also say $\phi_1, \phi_2, \ldots, \phi_n$ semantically entail $\psi$.

- Examples
  - $p \wedge q \models p$. For every valuation where $p \wedge q$ is T, $p$ must be T. Hence $p \wedge q \models p$.
  - $p \vee q \not\models q$. Consider the valuation $\{p \mapsto \mathrm{T}, q \mapsto \mathrm{F}\}$. We have $p \vee q$ is T but $q$ is F. Hence $p \vee q \not\models q$.
  - $\neg p, p \vee q \models q$. Consider any valuation where $\neg p$ and $p \vee q$ are T. Since $\neg p$ is T, $p$ must be F under the valuation. Since $p$ is F and $p \vee q$ is T, $q$ must be T under the valuation. Hence $\neg p, p \vee q \models q$.
- The validity of $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ is defined by syntactic calculus. $\phi_1, \phi_2, \ldots, \phi_n \models \psi$ is defined by truth tables. Do these two relations coincide?

# Soundness Theorem for Propositional Logic

## Theorem (Soundness)

*Let $\phi_1, \phi_2, \ldots, \phi_n$ and $\psi$ be propositional logic formulae. If $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ is valid, then $\phi_1, \phi_2, \ldots, \phi_n \models \psi$ holds.*

## Proof.

Consider the assertion $M(k)$:

"For all sequents $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi (n \geq 0)$ that have a proof of length $k$, then $\phi_1, \phi_2, \ldots, \phi_n \models \psi$ holds."

$k = 1$. The only possible proof is of the form

$$1 \quad \phi \quad \text{premise}$$

This is the proof of $\phi \vdash \phi$. For every valuation such that $\phi$ is T, $\phi$ must be T. That is, $\phi \models \phi$.

# Soundness Theorem for Propositional Logic

## Proof (cont'd).

Assume $M(i)$ for $i < k$. Consider a proof of the form

| | | |
|---|---|---|
| 1 | $\phi_1$ | premise |
| 2 | $\phi_2$ | premise |
| | $\vdots$ | |
| n | $\phi_n$ | premise |
| | $\vdots$ | |
| k | $\psi$ | justification |

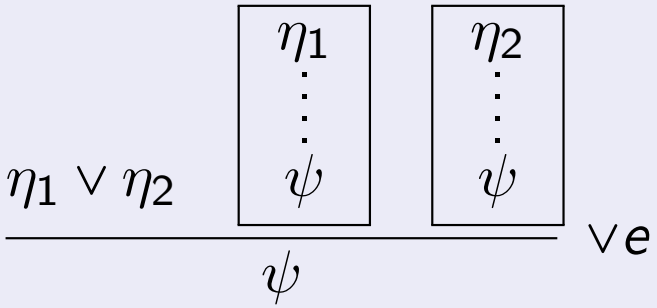
We have the following possible cases for justification:

i $\wedge i$. Then $\psi$ is $\psi_1 \wedge \psi_2$. In order to apply $\wedge i$, $\psi_1$ and $\psi_2$ must appear in the proof. That is, we have $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi_1$ and $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi_2$. By inductive hypothesis, $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi_1$ and $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi_2$. Hence $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi_1 \wedge \psi_2$ (Why?).

# Soundness Theorem for Propositional Logic

## Proof (cont'd).

ii $\vee e$. Recall the proof rule for $\vee e$:




In order to apply $\vee e$, $\eta_1 \vee \eta_2$ must appear in the proof. We have $\phi_1, \phi_2, \ldots, \phi_n \vdash \eta_1 \vee \eta_2$. By turning "assumptions" $\eta_1$ and $\eta_2$ to "premises," we obtain proofs for $\phi_1, \phi_2, \ldots, \phi_n, \eta_1 \vdash \psi$ and $\phi_1, \phi_2, \ldots, \phi_n, \eta_2 \vdash \psi$. By inductive hypothesis, $\phi_1, \phi_2, \ldots, \phi_n \vDash \eta_1 \vee \eta_2$, $\phi_1, \phi_2, \ldots, \phi_n, \eta_1 \vDash \psi$, and $\phi_1, \phi_2, \ldots, \phi_n, \eta_2 \vDash \psi$. Consider any valuation such that $\phi_1, \phi_2, \ldots, \phi_n$ evaluates to T. $\eta_1 \vee \eta_2$ must be T. If $\eta_1$ is T under the valuation, $\psi$ is also T (Why?). Similarly for $\eta_2$ is T. Thus $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$.

# Soundness Theorem for Propositional Logic

**Proof (cont'd).**

iii Other cases are similar. Prove the case of $\Longrightarrow e$ to see if you understand the proof.

□

- The soundness theorem shows that our calculus does not go wrong.
- If there is a proof of a sequent, then the conclusion must be true for all valuations where all premises are true.
- The theorem also allows us to show the non-existence of proofs.
- Given a sequent $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$, how do we prove there is no proof for the sequent?
  - Try to find a valuation where $\phi_1, \phi_2, \ldots, \phi_n$ are T but $\psi$ is F.

# Outline

# Completeness Theorem for Propositional Logic

- "$\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ is valid" and "$\phi_1, \phi_2, \ldots, \phi_n \models \psi$ holds" are very different.
  - "$\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ is valid" requires proof search (syntax);
  - "$\phi_1, \phi_2, \ldots, \phi_n \models \psi$ holds" requires a truth table (semantics).
- If "$\phi_1, \phi_2, \ldots, \phi_n \models \psi$ holds" implies "$\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ is valid," then our natural deduction proof system is complete.
- The natural deduction proof system is both sound and complete. That is

  $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ is valid iff $\phi_1, \phi_2, \ldots, \phi_n \models \psi$ holds.

# Completeness Theorem for Propositional Logic

- We will show the natural deduction proof system is complete.
- That is, if $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$ holds, then there is a natural deduction proof for the sequent $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$.
- Assume $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$. We proceed in three steps:
  1. $\vDash \phi_1 \Longrightarrow (\phi_2 \Longrightarrow (\ldots(\phi_n \Longrightarrow \psi)))$ holds;
  2. $\vdash \phi_1 \Longrightarrow (\phi_2 \Longrightarrow (\ldots(\phi_n \Longrightarrow \psi)))$ is valid;
  3. $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ is valid.

# Completeness Theorem for Propositional Logic (Step 1)

## Lemma

*If* $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$ *holds, then* $\vDash \phi_1 \Longrightarrow (\phi_2 \Longrightarrow (\ldots(\phi_n \Longrightarrow \psi)))$ *holds.*

## Proof.

Suppose $\vDash \phi_1 \Longrightarrow (\phi_2 \Longrightarrow (\ldots(\phi_n \Longrightarrow \psi)))$ does not hold. Then there is valuation where $\phi_1, \phi_2, \ldots, \phi_n$ is T but $\psi$ is F. A contradiction to $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$. □

## Definition

Let $\phi$ be a propositional logic formula. We say $\phi$ is a tautology if $\vDash \phi$.

- A tautology is a propositional logic formula that evaluates to T for all of its valuations.

# Completeness Theorem for Propositional Logic (Step 2)

- Our goal is to show the following theorem:

**Theorem**

*If $\vDash \eta$ holds, then $\vdash \eta$ is valid.*

- Similar to tautologies, we introduce the following definition:

**Definition**

Let $\phi$ be a propositional logic formula. We say $\phi$ is a theorem if $\vdash \phi$.

- Two types of theorems:
  - If $\vdash \phi$, $\phi$ is a theorem proved by the natural deduction proof system.
  - The soundness theorem for propositional logic is another type of theorem proved by mathematical reasoning (less formally).

# Completeness Theorem for Propositional Logic (Step 2)

## Proposition

*Let $\phi$ be a formula with propositional atoms $p_1, p_2, \ldots, p_n$. Let $l$ be a line in $\phi$'s truth table. For all $1 \leq i \leq n$, let $\hat{p}_i$ be $p_i$ if $p_i$ is $T$ in $l$; otherwise $\hat{p}_i$ is $\neg p_i$. Then*

1. *$\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \phi$ is valid if the entry for $\phi$ at $l$ is $T$;*
2. *$\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \neg\phi$ is valid if the entry for $\phi$ at $l$ is $F$.*

## Proof.

We prove by induction on the height of the parse tree of $\phi$.

- $\phi$ is a propositional atom $p$. Then $p \vdash p$ or $\neg p \vdash \neg p$ have one-line proof.
- $\phi$ is $\neg\phi_1$.
  - If $\phi$ is T at $l$. Then $\phi_1$ is F. By IH, $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \neg\phi_1 (\equiv \phi)$.
  - If $\phi$ is F at $l$. Then $\phi_1$ is T. By IH, $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \phi_1$. Using $\neg\neg i$, we have $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \neg\neg\phi_1 (\equiv \neg\phi)$.

# Completeness Theorem for Propositional Logic (Step 2)

## Proof (cont'd).

- $\phi$ is $\phi_1 \implies \phi_2$.
  - If $\phi$ is F at $I$, then $\phi_1$ is T and $\phi_2$ is F at $I$. By IH, $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \phi_1$ and $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \neg\phi_2$. Consider

| | | |
|---|---|---|
| 1 | $\phi_1 \implies \phi_2$ | assumption |
| | $\vdots$ | |
| i | $\phi_1$ | IH |
| i + 1 | $\phi_2$ | $\implies$ *e* i, 1 |
| | $\vdots$ | |
| j | $\neg\phi_2$ | IH |
| j + 1 | $\bot$ | ¬ e i+1, j |
| j + 2 | $\neg(\phi_1 \implies \phi_2)$ | ¬ i 1-(j+1) |

# Completeness Theorem for Propositional Logic (Step 2)

## Proof (cont'd).

- $\phi$ is $\phi_1 \Longrightarrow \phi_2$.
  - If $\phi$ is T at $I$, we have three subcases. Consider the case where $\phi_1$ and $\phi_2$ are F at $I$. Then

| | | |
|---|---|---|
| 1 | $\phi_1$ | assumption |
| | $\vdots$ | |
| i | $\neg\phi_1$ | IH |
| i + 1 | $\bot$ | $\neg$ e 1, i |
| i + 2 | $\phi_2$ | $\bot$ e (i+1) |
| i + 3 | $\phi_1 \Longrightarrow \phi_2$ | $\Longrightarrow$ i 1-(i+2) |

    The other two subcases are simple exercises.

# Completeness Theorem for Propositional Logic (Step 2)

## Proof (cont'd).

- $\phi$ is $\phi_1 \wedge \phi_2$.
  - If $\phi$ is T at $I$, then $\phi_1$ and $\phi_2$ are T at $I$. By IH, we have $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \phi_1$ and $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \phi_2$. Using $\wedge$ i, we have $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n \vdash \phi_1 \wedge \phi_2$.
  - If $\phi$ is F at $I$, there are three subcases. Consider the subcase where $\phi_1$ and $\phi_2$ are F at $I$. Then

| | | |
|---|---|---|
| 1 | $\phi_1 \wedge \phi_2$ | assumption |
| 2 | $\phi_1$ | $\wedge$ e$_1$ 1 |
| | $\vdots$ | |
| i | $\neg\phi_1$ | IH |
| i + 1 | $\bot$ | $\neg$ e 2, i |
| i + 2 | $\neg(\phi_1 \wedge \phi_2)$ | $\neg$ i 1-(i+1) |

    The other two subcases are simple exercises.

# Completeness Theorem for Propositional Logic (Step 2)

## Proof.

- $\phi$ is $\phi_1 \vee \phi_2$.
  - If $\phi$ is F at $I$, then $\phi_1$ and $\phi_2$ are F at $I$. Then

| | | | |
|---|---|---|---|
| 1 | $\phi_1 \vee \phi_2$ | assumption | ] |
| 2 | $\phi_1$ | assumption | ] \| |
| | $\vdots$ | | \| \| |
| i | $\neg\phi_1$ | IH | \| \| |
| i + 1 | $\bot$ | $\neg$ e 2, i | ] \| |
| i + 2 | $\phi_2$ | assumption | ] \| |
| | $\vdots$ | | \| \| |
| j | $\neg\phi_2$ | IH | \| \| |
| j + 1 | $\bot$ | $\neg$ e i+2, j | ] \| |
| j + 2 | $\bot$ | $\vee$ e 2-(i+1), (i+2)-(j+1) | ] |
| j + 3 | $\neg(\phi_1 \vee \phi_2)$ | $\neg$i 1-(j+2) | |

  - If $\phi$ is T at $I$, there are three subcases. All of them are simple exercises.

# Completeness Theorem for Propositional Logic (Step 2)

## Theorem

*If $\phi$ is a tautology, then $\phi$ is a theorem.*

## Proof.

Let $\phi$ have propositional atoms $p_1, p_2, \ldots, p_n$. Since $\phi$ is a tautology, each line in $\phi$'s truth table is T. By the above proposition, we have the following $2^n$ proofs for $\phi$:

$$\begin{array}{rcl} \neg p_1, \neg p_2, \ldots, \neg p_n & \vdash & \phi \\ p_1, \neg p_2, \ldots, \neg p_n & \vdash & \phi \\ \neg p_1, p_2, \ldots, \neg p_n & \vdash & \phi \\ & \vdots & \\ p_1, p_2, \ldots, p_n & \vdash & \phi \end{array}$$

We apply the rule LEM and the $\vee$e rule to obtain a proof for $\vdash \phi$. (See the following example.) $\square$

# Completeness Theorem for Propositional Logic (Step 2)

## Example

Observe that $\vDash p \Longrightarrow (q \Longrightarrow p)$. Prove $\vdash p \Longrightarrow (q \Longrightarrow p)$.

## Proof.

| | | |
|---|---|---|
| 1 | $p \vee \neg p$ | LEM |
| 2 | $p$ | assumption |
| 3 | $q \vee \neg q$ | LEM |
| 4 | $q$ | assumption |
| | $\vdots$ | |
| i | $p \Longrightarrow (q \Longrightarrow p)$ | $p, q \vdash p \Longrightarrow (q \Longrightarrow p)$ |
| i + 1 | $\neg q$ | assumption |
| | $\vdots$ | |
| j | $p \Longrightarrow (q \Longrightarrow p)$ | $p, \neg q \vdash p \Longrightarrow (q \Longrightarrow p)$ |
| j + 1 | $p \Longrightarrow (q \Longrightarrow p)$ | $\vee$e 3, 4-i, (i+1)-j |
| j + 2 | $\neg p$ | assumption |
| j + 3 | $q \vee \neg q$ | LEM |
| j + 4 | $q$ | assumption |
| | $\vdots$ | |
| k | $p \Longrightarrow (q \Longrightarrow p)$ | $\neg p, q \vdash p \Longrightarrow (q \Longrightarrow p)$ |
| k + 1 | $\neg q$ | assumption |
| | $\vdots$ | |
| l | $p \Longrightarrow (q \Longrightarrow p)$ | $\neg p, \neg q \vdash p \Longrightarrow (q \Longrightarrow p)$ |
| l + 1 | $p \Longrightarrow (q \Longrightarrow p)$ | $\vee$e (j+3), (j+4)-k, (k+1)-l |
| l + 2 | $p \Longrightarrow (q \Longrightarrow p)$ | $\vee$e 1, 2-(j+1), (j+2)-(l+1) |

□

# Completeness Theorem for Propositional Logic (Step 3)

## Lemma

*If* $\phi_1 \Longrightarrow (\phi_2 \Longrightarrow (\cdots(\phi_n \Longrightarrow \psi)))$ *is a theorem, then* $\phi_1, \phi_2, \ldots, \phi_n \vdash \psi$ *is valid.*

## Proof.

Consider

| | | |
|---|---|---|
| 1 | $\phi_1$ | premise |
| 2 | $\phi_2$ | premise |
| | $\vdots$ | |
| n | $\phi_n$ | premise |
| | $\vdots$ | |
| i | $\phi_1 \Longrightarrow (\phi_2 \Longrightarrow (\cdots(\phi_n \Longrightarrow \psi)))$ | theorem |
| i + 1 | $\phi_2 \Longrightarrow (\cdots(\phi_n \Longrightarrow \psi))$ | $\Longrightarrow$ e 1, i |
| i + 2 | $\phi_3 \Longrightarrow (\cdots(\phi_n \Longrightarrow \psi))$ | $\Longrightarrow$ e 2, (i+1) |
| | $\vdots$ | |
| i + n - 1 | $\phi_n \Longrightarrow \psi$ | $\Longrightarrow$ e (n-1), (i+n-2) |
| i + n | $\psi$ | $\Longrightarrow$ e n, (i+n-1) |

# Outline

# Semantically Equivalence and Validity

- Consider two formulae $\phi_1 \wedge \phi_2$ and $\phi_2 \wedge \phi_1$.
- Intuitively, $\phi_1 \wedge \phi_2$ and $\phi_2 \wedge \phi_1$ should have the same "meaning."
- More formally, two formulae $\phi$ and $\psi$ have the same meaning if their truth tables coincide.

**Definition**

Let $\phi$ and $\psi$ be propositional logic formulae. $\phi$ and $\psi$ are semantically equivalent (written $\phi \equiv \psi$) if both $\phi \models \psi$ and $\psi \models \phi$ hold.

- Examples

$$p \Longrightarrow q \equiv \neg q \Longrightarrow \neg p \qquad p \Longrightarrow q \equiv \neg p \vee q$$
$$p \wedge q \Longrightarrow p \equiv r \vee \neg r \qquad p \wedge q \Longrightarrow r \equiv p \Longrightarrow (q \Longrightarrow r)$$

- A formula $\phi$ is valid if it is a tautology.

**Definition**

Let $\phi$ be a propositional logic formula. $\phi$ is valid if $\models \phi$.

# Semantic Entailment and Validity

## Lemma

*Let* $\phi_1, \phi_2, \ldots, \phi_n, \psi$ *be propositional logic formulae.* $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$ *iff* $\vDash \phi_1 \Longrightarrow (\phi_2 \Longrightarrow \cdots \Longrightarrow (\phi_n \Longrightarrow \psi))$.

## Proof.

Suppose $\vDash \phi_1 \Longrightarrow (\phi_2 \Longrightarrow \cdots \Longrightarrow (\phi_n \Longrightarrow \psi))$ Consider any valuation. If $\phi_1, \phi_2, \ldots, \phi_n$ evaluate to T under the valuation, $\phi$ must evaluate to T since $\vDash \phi_1 \Longrightarrow (\phi_2 \Longrightarrow \cdots \Longrightarrow (\phi_n \Longrightarrow \psi))$. Hence $\phi_1, \phi_2, \ldots, \phi_n \vDash \psi$.
The other direction is proved in Step 1 of the completeness theorem. □

# Conjunctive Normal Form (CNF)

## Definition

A literal $L$ is either an atom $p$ or its negation $\neg p$. A clause $D$ is a disjunction of literals. A formula $C$ is in conjunctive normal form (CNF) if it is a conjunction of clauses.

$$\begin{array}{rcl} L & ::= & p \mid \neg p \\ D & ::= & L \mid L \vee D \\ C & ::= & D \mid D \wedge C \end{array}$$

- Examples: $(\neg q \vee p \vee r) \wedge (\neg p \vee r) \wedge q$, $(p \vee r) \wedge (\neg p \vee r) \wedge (p \vee \neg r)$

# Validity of CNF Formulae

## Lemma

*A clause $L_1 \vee L_2 \vee \cdots \vee L_m$ is valid iff there is a propositional atom $p$ such that $L_i$ is $p$ and $L_j$ is $\neg p$ for some $1 \leq i, j \leq m$.*

## Proof.

Without loss of generality, assume $L_1 = p$ and $L_2 = \neg p$. Then $p \vee \neg p \vee L_3 \vee \cdots \vee L_m$ evaluates to T for any valuation. The clause is valid. Conversely, consider the valuation where all literals evaluate to F. This is possible since every literal $L_i$ has no negation in the clause. The clause evaluates to F under the valuation. □

- Examples:
  - $p \vee q \vee q \vee \neg p \vee r$ is valid;
  - $p \vee \neg q \vee r \vee \neg q$ is not valid (consider $\{p \mapsto \mathrm{F}, q \mapsto \mathrm{T}, r \mapsto \mathrm{F}\}$).
- For any propositional logic formula $\phi$ in CNF, the validity of $\phi$ can be checked in linear time.

# Satisfiability of CNF Formulae

## Definition

Let $\phi$ be a propositional logic formula. $\phi$ is satisfiable if it evaluates to T under some valuation.

- Example: $p \vee q \implies p$ is satisfiable (consider $\{p \mapsto \mathsf{T}, q \mapsto \mathsf{T}\}$); it is not valid (consider $\{p \mapsto \mathsf{F}, q \mapsto \mathsf{T}\}$).

## Proposition

*Let $\phi$ be a propositional logic formula. $\phi$ is satisfiable iff $\neg\phi$ is not valid.*

## Proof.

Suppose $\phi$ evaluates to T under a valuation. Then $\neg\phi$ evaluates to F under the valuation. $\neg\phi$ is not valid.
Conversely, suppose $\neg\phi$ is not valid. Hence $\neg\phi$ evaluates to F under a valuation. Thus $\phi$ evaluates to T under the valuation. $\phi$ is satisfiable. □

# From Truth Tables to Conjunctive Normal Form

- Suppose we have the truth table for a formula $\phi$ with propositional atoms $p_1, p_2, \ldots, p_n$.
- For each line $l$ where $\phi$ evaluates to F, construct a clause $\psi_l$ as follows.
  - $\psi_l = L_{l,1} \vee L_{l,2} \vee \cdots \vee L_{l,n}$ where $L_{l,j} = \neg p_j$ if $p_j$ is T at line $l$; otherwise $L_{l,j} = p_j$.
- Then $\phi \equiv \psi_1 \wedge \psi_2 \wedge \cdots \psi_m$ where $\psi_l$'s are contructed for every line evaluating $\phi$ to F.
- Observe that $\psi_1 \wedge \psi_2 \wedge \cdots \psi_m$ is F iff $\psi_l$ is F for some $1 \leq l \leq m$. $\psi_l = L_{l,1} \vee L_{l,2} \vee \cdots \vee L_{l,n}$ is F iff $L_{l,j}$ is F for every $1 \leq j \leq n$. $L_{l,j}$ is F iff $p_j$ has its truth value at line $l$.
- In other words, $\psi_1 \wedge \psi_2 \wedge \cdots \psi_m$ is F under a valuation iff the valuation evaluates $\phi$ to F in $\phi$'s truth table.

# From Truth Tables to Conjunctive Normal Form

## Example

Translate $p \vee q \implies q \wedge \neg r$ into CNF.

## Proof.

| $p$ | $q$ | $r$ | $p \vee q \implies q \wedge \neg r$ |
|---|---|---|---|
| F | F | F | T |
| F | F | T | T |
| F | T | F | T |
| F | T | T | F |

| $p$ | $q$ | $r$ | $p \vee q \implies q \wedge \neg r$ |
|---|---|---|---|
| T | F | F | F |
| T | F | T | F |
| T | T | F | T |
| T | T | T | F |

| $p$ | $q$ | $r$ | $\psi_I$ |
|---|---|---|---|
| F | T | T | $p \vee \neg q \vee \neg r$ |
| T | F | T | $\neg p \vee q \vee \neg r$ |

| $p$ | $q$ | $r$ | $\psi_I$ |
|---|---|---|---|
| T | F | F | $\neg p \vee q \vee r$ |
| T | T | T | $\neg p \vee \neg q \vee \neg r$ |

$p \vee q \implies q \wedge \neg r \equiv (p \vee \neg q \vee \neg r) \wedge (\neg p \vee q \vee r) \wedge (\neg p \vee q \vee \neg r) \wedge (\neg p \vee \neg q \vee \neg r)$. □

# Outline

# Validity Checking

- Given a propositional logic formula in conjunctive normal form, we can check the validity of the formula in linear time.
- Recall that a formula is valid iff it is a theorem.
- If we can translate any propositional logic formula into conjunctive normal form, we can check the validity of the formula!
- We know how to translate any logic formula to conjunctive normal form by its truth table.
  - This is not satisfactory. If we have to construct its truth table, we can check validity already.
- We will give an algorithm $\texttt{CNF}(\phi)$ to convert any propositional logic formula into conjunctive normal form without building its truth table.

# From Formula to Conjunctive Normal Form

- Any propositional logic formula can be transformed to conjunctive normal form by the following equivalences:

$$\phi \implies \psi \equiv \neg\phi \vee \psi$$
$$\neg(\phi \wedge \psi) \equiv \neg\phi \vee \neg\psi \qquad \neg(\phi \vee \psi) \equiv \neg\phi \wedge \neg\psi$$
$$\phi \wedge (\psi_1 \vee \psi_2) \equiv (\phi \wedge \psi_1) \vee (\phi \wedge \psi_2)$$
$$\phi \vee (\psi_1 \wedge \psi_2) \equiv (\phi \vee \psi_1) \wedge (\phi \vee \psi_2)$$

- The algorithm $\texttt{CNF}(\phi)$ hence consists of three steps:
  - Remove every implication ( $\implies$ ) from $\phi$ (Algorithm $\texttt{IMPL\_FREE}(\phi)$);
  - Push every negation ($\neg$) to literals (Algorithm $\texttt{NNF}(\phi)$);
  - Apply law of distribution (Algorithm $\texttt{CNF}(\phi)$).

# Algorithm IMPL_FREE($\phi$)

**Input:** $\phi$ : a logic formula
**Output:** $\phi'$ : all implications ( $\Longrightarrow$ ) in $\phi'$ are removed and $\phi' \equiv \phi$
**switch** $\phi$ **do**
    **case** $\phi$ is a literal: **do return** $\phi$;
    **case** $\phi$ is $\neg\phi_1$: **do return** $\neg\texttt{IMPL\_FREE}(\phi_1)$;
    **case** $\phi$ is $\phi_1 \wedge \phi_2$: **do return** $\texttt{IMPL\_FREE}(\phi_1) \wedge \texttt{IMPL\_FREE}(\phi_2)$;
    **case** $\phi$ is $\phi_1 \vee \phi_2$: **do return** $\texttt{IMPL\_FREE}(\phi_1) \vee \texttt{IMPL\_FREE}(\phi_2)$;
    **case** $\phi$ is $\phi_1 \Longrightarrow \phi_2$: **do return** $\texttt{IMPL\_FREE}(\neg\phi_1 \vee \phi_2)$;
    **otherwise do** assert(0);

**Algorithm 1:** IMPL_FREE($\phi$)

# Algorithm $\texttt{NNF}(\phi)$

**Input:** $\phi$ : a logic formula without implication ( $\Longrightarrow$ )
**Output:** $\phi'$ : only propositional atoms in $\phi'$ are negated and $\phi' \equiv \phi$
**switch** $\phi$ **do**
- **case** $\phi$ is a literal: **do return** $\phi$;
- **case** $\phi$ is $\neg\neg\phi_1$: **do return** $\texttt{NNF}(\phi_1)$;
- **case** $\phi$ is $\phi_1 \wedge \phi_2$: **do return** $\texttt{NNF}(\phi_1) \wedge \texttt{NNF}(\phi_2)$;
- **case** $\phi$ is $\phi_1 \vee \phi_2$: **do return** $\texttt{NNF}(\phi_1) \vee \texttt{NNF}(\phi_2)$;
- **case** $\phi$ is $\neg(\phi_1 \wedge \phi_2)$: **do return** $\texttt{NNF}(\neg\phi_1 \vee \neg\phi_2)$;
- **case** $\phi$ is $\neg(\phi_1 \vee \phi_2)$: **do return** $\texttt{NNF}(\neg\phi_1 \wedge \neg\phi_2)$;
- **otherwise do** $\texttt{assert}(0)$;

**Algorithm 2:** $\texttt{NNF}(\phi)$

## Definition

Let $\phi$ be a propositional logic formula. If only propositional atoms in $\phi$ are negated, $\phi$ is in negation normal form.

# Algorithm $\texttt{CNF}(\phi)$

**Input:** $\phi$ : an NNF formula without implication ( $\Longrightarrow$ )
**Output:** $\phi'$ : $\phi'$ is in CNF and $\phi' \equiv \phi$
**switch** $\phi$ **do**
  **case** $\phi$ is a literal: **do return** $\phi$;
  **case** $\phi$ is $\phi_1 \wedge \phi_2$: **do return** $\texttt{CNF}(\phi_1) \wedge \texttt{CNF}(\phi_2)$;
  **case** $\phi$ is $\phi_1 \vee \phi_2$: **do return** $\texttt{DISTR}(\texttt{CNF}(\phi_1), \texttt{CNF}(\phi_2))$;

**Algorithm 3:** $\texttt{CNF}(\phi)$

**Input:** $\eta_1, \eta_2$ : $\eta_1, \eta_2$ are in CNF
**Output:** $\phi'$ : $\phi'$ is in CNF and $\phi' \equiv \eta_1 \vee \eta_2$
**if** $\eta_1$ is $\eta_{11} \wedge \eta_{12}$ **then return** $\texttt{DISTR}(\eta_{11}, \eta_2) \wedge \texttt{DISTR}(\eta_{12}, \eta_2)$ ;
**else if** $\eta_2$ is $\eta_{21} \wedge \eta_{22}$ **then return** $\texttt{DISTR}(\eta_1, \eta_{21}) \wedge \texttt{DISTR}(\eta_1, \eta_{22})$ ;
**else return** $\eta_1 \vee \eta_2$ ;

**Algorithm 4:** $\texttt{DISTR}(\eta_1, \eta_2)$

# Satisfiability of Propositional Logic Formulae

- Let $\phi$ be a propositional logic formula. Consider the following algorithm for checking its satisfiability.
  1. Compute a CNF formula $\psi$ such that $\psi \equiv \neg\phi$.
  2. Check the validity of $\psi$.
  3. Return "$\phi$ is satisfiable" if $\psi$ is not valid; Return "$\phi$ is not satisfiable" if $\psi$ is valid.
- Recall that satisfiability of propositional logic formulae is an NP-complete problem.
- Is the above algorithm in polynomial time? Why?

# Outline

# Horn Clauses

- Given a propositional logic formula in CNF, it is easy to check its validity; it is "hard" to check its satisfiability.
- We will consider a subclass of CNF formulae whose satisfiability can be checked efficiently.

## Definition

A Horn formula is a propositional logic formula $\phi$ of the following form:

$$\begin{array}{lcl} P & ::= & \bot \mid \top \mid p \\ A & ::= & P \mid P \wedge A \\ C & ::= & A \Longrightarrow P \\ H & ::= & C \mid C \wedge H. \end{array}$$

A clause of the form $C$ is called a Horn clause.

- Example: $(p \wedge q \wedge s \Longrightarrow \bot) \wedge (q \wedge r \Longrightarrow p) \wedge (\top \Longrightarrow s)$
- Nonexample: $(p \wedge \neg q \wedge s \Longrightarrow \bot) \wedge (q \wedge r \Longrightarrow p \wedge s) \wedge (p \vee r \Longrightarrow s)$

# Satisfiability of Horn Formulae

- Consider a Horn clause $P_1 \wedge P_2 \wedge \cdots \wedge P_n \implies Q$.
- If $P_1, P_2, \ldots, P_n$ are assigned to T, then $Q$ must be T; otherwise, $Q$ can be an arbitrary truth value.
- We hence have the following (informal) algorithm:
  1. Mark $\top$ if it occurs in the Horn formula $\phi$;
  2. If there is a Horn clause $P_1 \wedge P_2 \wedge \cdots \wedge P_n \implies Q$ in $\phi$ such that all $P_j$ for $1 \leq j \leq n$ are marked, mark $Q$;
  3. If $\bot$ is marked, print "The Horn formula $\phi$ is unsatisfiable."
  4. Print "The Horn formula $\phi$ is satisfiable."

# Algorithm Horn($\phi$)

**Input:** $\phi$: $\phi$ is a Horn formula
**Output:** "unsatisfiable" if $\phi$ is unsatisfiable; otherwise "satisfiable."
mark all occurrences of $\top$ in $\phi$;
**while** there is a Horn clause $P_1 \wedge P_2 \wedge \cdots \wedge P_n \Longrightarrow Q$ in $\phi$ such that $P_j$ are all marked but $Q$ is not **do**
    mark $Q$;
**if** $\bot$ is marked **then return** "unsatisfiable" ;
**else return** "satisfiable" ;

**Algorithm 5:** Horn($\phi$)

# Satisfiability of Horn Formulae

## Theorem

*Let $\phi$ be a Horn formula with $n$ propositional atoms.* `Horn`$(\phi)$ *runs at most $n+1$ iterations and decides the satisfiability of $\phi$ correctly.*

## Proof.

At each iteration, an unmarked atom will be marked. Since there are $n$ atoms, there are at most $n+1$ iterations.

By induction on the number of iterations, we show that "all marked $P$ are true for all valuations where $\phi$ evaluates to T." At iteration 0 (before entering the loop), only $\top$ are marked. Clearly, $\top$ must be true for any valuation. At iteration $k+1$, consider a Horn clause $P_1 \wedge P_2 \wedge \cdots \wedge P_n \implies Q$ where $P_1, P_2, \ldots, P_n$ are marked but not $Q$. For a valuation $\nu$ where $\phi$ evaluates to T, $P_1 \wedge P_2 \wedge \cdots \wedge P_n \implies Q$ must evaluate to T. Since $P_1, P_2, \ldots, P_n$ are true in $\nu$ (by IH), $Q$ must be true in $\nu$.

# Satisfiability of Horn Formulae

## Proof (cont'd).

We now prove $\texttt{Horn}(\phi)$ answers correctly. When $\texttt{Horn}(\phi)$ returns "unsatisfiable," there is a Horn clause $P_1 \wedge P_2 \wedge \cdots \wedge P_n \implies \bot$ where $P_1, P_2, \ldots, P_n$ are all marked. Suppose $\nu$ is a valuation where $\phi$ evaluates to T. Then $P_1, P_2, \ldots, P_n$ must be true in $\nu$. Hence $P_1 \wedge P_2 \wedge \cdots \wedge P_n \implies \bot$ evaluates to F. $\phi$ cannot evaluate to T under $\nu$. A contradiction.

When $\texttt{Horn}(\phi)$ returns "satisfiable," define a valuation $\nu$ where all marked propositional atoms are assigned to T and all unmarked atoms are F. We claim $\phi$ evaluates to T in $\nu$. Suppose not. There is a Horn clause $P_1 \wedge P_2 \wedge \cdots P_n \implies Q$ in $\phi$ which evaluates to F under $\nu$. That is, $P_1, P_2, \ldots, P_n$ are T but $Q$ is F under $\nu$. By the definition of $\nu$, $P_1, P_2, \ldots, P_n$ are marked by the algorithm. Hence $Q$ must also be marked by the algorithm. $Q$ cannot be F in $\nu$. A contradiction. □

# Outline

# SAT Solvers

- The satisfiability problem for propositional logic is to decide whether a propositional logic formula $\phi$ is satisfiable.
  - The satisfiability problem for propositional logic is an NP-complete problem (Cook's Theorem).
- Many SAT solvers are available for the problem.
- We will discuss algorithms for the satisfiability problem.
- For this topic, most materials are copied or modified from Prof. Sharad Malik's and Prof. Chung-Yang Huang's lecture notes.

# Equisatisfiable Propositional Logic Formulae

- Let $\phi$ and $\psi$ be propositional logic formulae.
- $\phi$ and $\psi$ are equisatisfiable if

  $\phi$ is satisfiable if and only if $\psi$ is satisfiable.

- What is the difference between semantic equivalence and equisatisfiability?

# Tseitin Transformation

## Theorem

*For every propositional logic formula $\phi$, there is a propositional logic formula $\psi$ in CNF such that $\phi$ and $\psi$ are equisatisfiable.*

## Proof.

For every subformula $\alpha$ of $\phi$, $x_\alpha$ is $p$ if $\alpha$ is the atomic proposition $p$; $x_\alpha$ is $P_\alpha$ otherwise. For every non-atomic subformula $\alpha$, define $C_\alpha$ as follows.

| $\alpha$ | $C_\alpha$ | Remark |
|---|---|---|
| $\neg\beta$ | $(x_\alpha \vee x_\beta) \wedge (\neg x_\alpha \vee \neg x_\beta)$ | $x_\alpha \Leftrightarrow \neg x_\beta$ |
| $\beta_0 \vee \beta_1$ | $(x_\alpha \vee \neg x_{\beta_0}) \wedge (x_\alpha \vee \neg x_{\beta_1}) \wedge (\neg x_\alpha \vee x_{\beta_0} \vee x_{\beta_1})$ | $x_\alpha \Leftrightarrow x_{\beta_0} \vee x_{\beta_1}$ |
| $\beta_0 \wedge \beta_1$ | $(\neg x_\alpha \vee x_{\beta_0}) \wedge (\neg x_\alpha \vee x_{\beta_1}) \wedge (x_\alpha \vee \neg x_{\beta_0} \vee \neg x_{\beta_1})$ | $x_\alpha \Leftrightarrow x_{\beta_0} \wedge x_{\beta_1}$ |

Let $\psi = x_\phi \wedge \bigwedge\{C_\alpha : \alpha \text{ is a non-atomic subformula of } \phi\}$. Then $\phi$ and $\psi$ are equisatisfiable. □

# Example

## Example

Transform $\phi = \neg(p \wedge \neg q)$ into an equisatisfiable propositional logic formula in CNF.

## Proof.

$\neg q, p \wedge \neg q, \phi$ are non-atomic subformulae of $\phi$.

$$\begin{aligned} C_{\neg q} &\stackrel{\triangle}{=} (P_{\neg q} \vee q) \wedge (\neg P_{\neg q} \vee \neg q) \\ C_{p \wedge \neg q} &\stackrel{\triangle}{=} (\neg P_{p \wedge \neg q} \vee p) \wedge (\neg P_{p \wedge \neg q} \vee P_{\neg q}) \wedge (P_{p \wedge \neg q} \vee \neg p \vee \neg P_{\neg q}) \\ C_{\phi} &\stackrel{\triangle}{=} (P_{\phi} \vee P_{p \wedge \neg q}) \wedge (\neg P_{\phi} \vee \neg P_{p \wedge \neg q}) \end{aligned}$$

$\phi$ and $P_{\phi} \wedge C_{\phi} \wedge C_{p \wedge \neg q} \wedge C_{\neg q} =$
$P_{\phi} \wedge (P_{\phi} \vee P_{p \wedge \neg q}) \wedge (\neg P_{\phi} \vee \neg P_{p \wedge \neg q}) \wedge (\neg P_{p \wedge \neg q} \vee p) \wedge (\neg P_{p \wedge \neg q} \vee P_{\neg q}) \wedge$
$(P_{p \wedge \neg q} \vee \neg p \vee \neg P_{\neg q}) \wedge (P_{\neg q} \vee q) \wedge (\neg P_{\neg q} \vee \neg q)$ are equisatisfiable. □

# Example

## Example

Transform $\phi = p \vee \neg(q \wedge (r \vee \neg s))$ into an equisatisfiable propositional logic formula in CNF.

## Proof.

$\neg s, r \vee \neg s, q \wedge (r \vee \neg s), \neg(q \wedge (r \vee \neg s)), \phi$ are non-atomic subformulae of $\phi$.

$$
\begin{aligned}
C_{\neg s} &\stackrel{\triangle}{=} (P_{\neg s} \vee s) \wedge (\neg P_{\neg s} \vee \neg s) \\
C_{r \vee \neg s} &\stackrel{\triangle}{=} (P_{r \vee \neg s} \vee \neg r) \wedge (P_{r \vee \neg s} \vee \neg P_{\neg s}) \wedge (\neg P_{r \vee \neg s} \vee r \vee P_{\neg s}) \\
C_{q \wedge (r \vee \neg s)} &\stackrel{\triangle}{=} (\neg P_{q \wedge (r \vee \neg s)} \vee q) \wedge (\neg P_{q \wedge (r \vee \neg s)} \vee P_{r \vee \neg s}) \wedge \\
&\quad (P_{q \wedge (r \vee \neg s)} \vee \neg q \vee \neg P_{r \vee \neg s}) \\
C_{\neg(q \wedge (r \vee \neg s))} &\stackrel{\triangle}{=} (P_{\neg(q \wedge (r \vee \neg s))} \vee P_{q \wedge (r \vee \neg s)}) \wedge (\neg P_{\neg(q \wedge (r \vee \neg s))} \vee \neg P_{q \wedge (r \vee \neg s)}) \\
C_{\phi} &\stackrel{\triangle}{=} (P_{\phi} \vee \neg p) \wedge (P_{\phi} \vee \neg P_{\neg(q \wedge (r \vee \neg s))}) \wedge (\neg P_{\phi} \vee p \vee P_{\neg(q \wedge (r \vee \neg s))})
\end{aligned}
$$

Then $\phi$ and $P_{\phi} \wedge C_{\phi} \wedge C_{\neg(q \wedge (r \vee \neg s))} \wedge C_{q \wedge (r \vee \neg s)} \wedge C_{r \vee \neg s} \wedge C_{\neg s}$ is equisatisfiable. □

# Properties of Tseitin Transformation

- Let $\phi$ be a propositional logic formula.
- The size $|\phi|$ of $\phi$ is the number of symbols (atomic propositions, $\wedge$, $\vee$, $\neg$) in $\phi$.
  - Parentheses ("(" and ")") do not count.
  - $|\phi|$ is the number of nodes in the parsing tree of $\phi$.
- Tseitin Transformation of $\phi$ has
  - $|\phi|$ atomic propositions;
  - Each $C_\alpha$ has at most 3 clauses;
  - Each clause of $C_\alpha$ has at most 3 literals.
- Let
  $\phi = (p_{11} \wedge p_{12} \wedge \cdots p_{1m_1}) \vee (p_{21} \wedge p_{22} \wedge \cdots p_{2m_2}) \vee \cdots \vee (p_{n1} \wedge p_{n2} \wedge \cdots p_{nm_n})$.
  We obtain a semantic equivalent propositional logic formula $\psi$ by distributive law. What is the size of $\psi$?

# DIMACS SAT Format I

- DIMACS SAT format is a standard text format for CNF formulae.
- Most SAT solvers accept a simplified version of DIMACS SAT format.
- Here is the example from the SAT Competition home page.

```
c
c start with comments
c
c
p cnf 5 3
1 -5 4 0
-1 5 3 4 0
-3 -4 0
```

# DIMACS SAT Format II

- From the SAT Competition home page:
  - The file can start with comments, that is lines begining with the character c.
  - Right after the comments, there is the line `p cnf` *nbvar nbclauses* indicating that the instance is in CNF format; *nbvar* is the exact number of variables appearing in the file; *nbclauses* is the exact number of clauses contained in the file.
  - Then the clauses follow. Each clause is a sequence of distinct non-null numbers between -*nbvar* and *nbvar* ending with 0 on the same line; it cannot contain the opposite literals $i$ and $-i$ simultaneously. Positive numbers denote the corresponding variables. Negative numbers denote the negations of the corresponding variables.
- Following DIMACS SAT format, we will represent a propositional logic formula in CNF by a set of clauses.
  - $(\neg a \vee b \vee \neg c) \wedge (\neg b \vee d) \wedge (a \vee c \vee \neg d)$ is represented as

$$(\neg a \vee b \vee \neg c) \quad (\neg b \vee d) \quad (a \vee c \vee \neg d)$$

# SAT Algorithms

- Davis, Putnam, 1960
  - Explicit resolution based.
  - May explode in memory.
- Davis, Logemann, Loveland, (DLL) 1962
  - Search based.
  - Most successful, basis for almost all modern SAT solvers.
  - Learning and non-chronological backtracking, 1996.
- Stålmarcks algorithm, 1980s
  - Proprietary algorithm. Patented.
  - Commercial versions available
- Stochastic Methods, 1992
  - Unable to prove unsatisfiability, but may find solutions for a satisfying problem quickly.
  - Local search and hill climbing

# Resolution

- Consider the proof rule:

$$\frac{l_0 \vee l_1 \vee \cdots \vee l_m \vee k \quad \overline{k} \vee l'_0 \vee l'_1 \vee \cdots \vee l'_n}{l_0 \vee l_1 \vee \cdots \vee l_m \vee l'_0 \vee l'_1 \vee \cdots \vee l'_n} \; \textit{resolution}$$

  - $k$ and $\overline{k}$ are complementary literals.
  - We assume $\vee$ is commutative.
- For instance, we obtain $(a \vee \neg b \vee \neg d \vee f)$ from $(a \vee \neg b \vee \neg c)$ and $(\neg d \vee c \vee f)$ by resolution.

# Davis Putnam Algorithm

- Select a variable for resolution iteratively.
- Consider the following set of clauses:

$$\begin{array}{llll} (a \vee b \vee c) & (b \vee \neg c \vee f) & (\neg b \vee e) & \\ \hline (a \vee c \vee e) & (\neg c \vee e \vee f) & & \\ \hline (a \vee e \vee f) & & & \end{array}$$

SAT!

- Consider the following set of clauses:

$$\begin{array}{llll} (a \vee b) & (a \vee \neg b) & (\neg a \vee c) & (\neg a \vee \neg c) \\ \hline (a) & (\neg a \vee c) & (\neg a \vee \neg c) & \\ \hline (c) & (\neg c) & & \\ \hline () & & & \end{array}$$

UNSAT!

# Basic DLL I

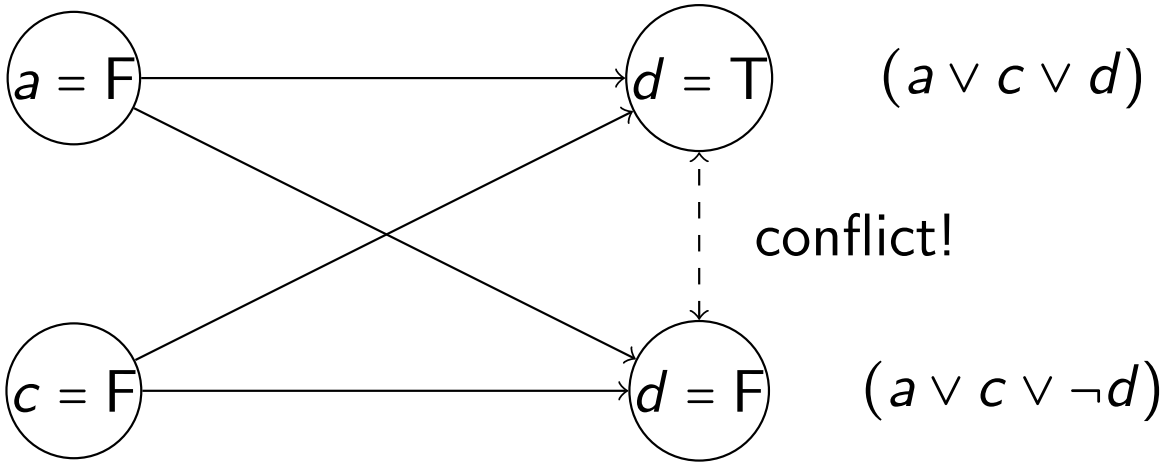
- Consider the following set of clauses:

$$(\neg a \vee b \vee c) \quad (a \vee c \vee d) \quad (a \vee c \vee \neg d) \quad (a \vee \neg c \vee d)$$
$$(a \vee \neg c \vee \neg d) \quad (\neg b \vee \neg c \vee d) \quad (\neg a \vee b \vee \neg c) \quad (\neg a \vee \neg b \vee c)$$

- We perform depth first search:

$$a = \mathrm{F} \quad b = \mathrm{F} \quad c = \mathrm{F}$$

and obtain a conflict:




  - This is an implication graph.

# Basic DLL II

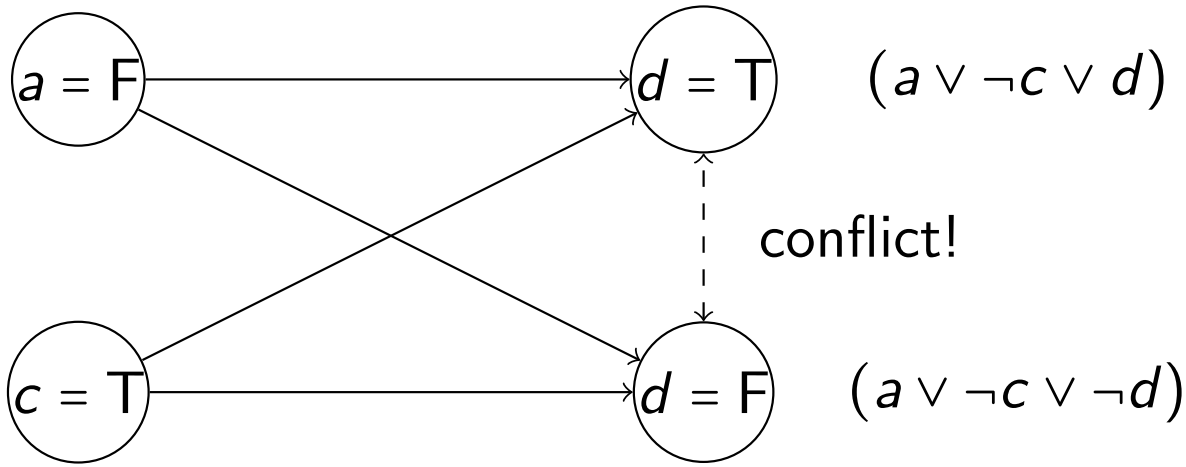
- Consider the following set of clauses:

$$(\neg a \vee b \vee c) \quad (a \vee c \vee d) \quad (a \vee c \vee \neg d) \quad (a \vee \neg c \vee d)$$
$$(a \vee \neg c \vee \neg d) \quad (\neg b \vee \neg c \vee d) \quad (\neg a \vee b \vee \neg c) \quad (\neg a \vee \neg b \vee c)$$

- Backtrack!

$$a = \mathrm{F} \quad b = \mathrm{F} \quad \cancel{c = \mathrm{F}}$$
$$c = \mathrm{T}(\text{forced})$$

and obtain a conflict:



# Basic DLL III

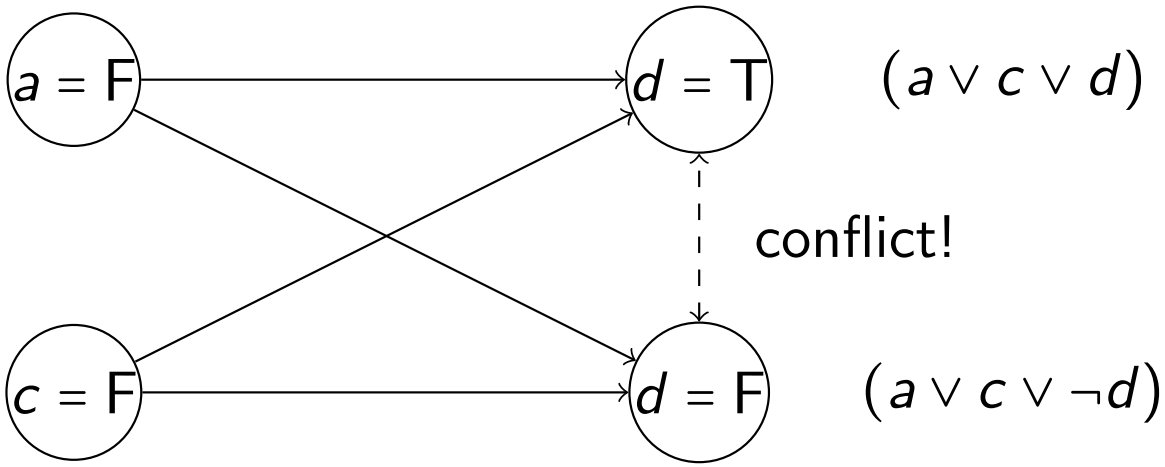
- Consider the following set of clauses:

$$(\neg a \vee b \vee c) \quad (a \vee c \vee d) \quad (a \vee c \vee \neg d) \quad (a \vee \neg c \vee d)$$
$$(a \vee \neg c \vee \neg d) \quad (\neg b \vee \neg c \vee d) \quad (\neg a \vee b \vee \neg c) \quad (\neg a \vee \neg b \vee c)$$

- Backtrack!

$$a = \mathsf{F} \quad \cancel{b = \mathsf{F}}$$
$$b = \mathsf{T}(\text{forced}) \quad c = \mathsf{F}$$

and obtain a conflict:



# Basic DLL IV

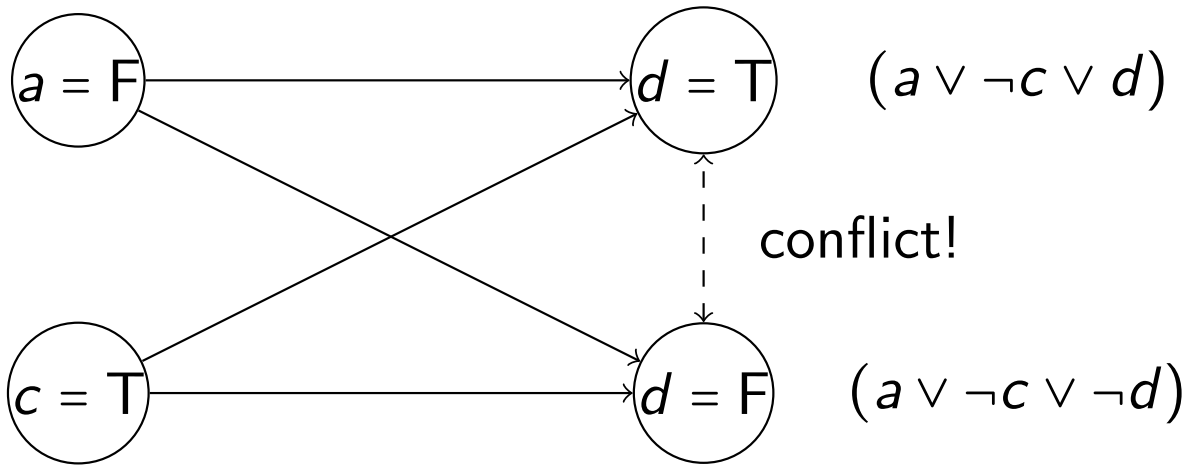
- Consider the following set of clauses:

$$(\neg a \vee b \vee c) \quad (a \vee c \vee d) \quad (a \vee c \vee \neg d) \quad (a \vee \neg c \vee d)$$
$$(a \vee \neg c \vee \neg d) \quad (\neg b \vee \neg c \vee d) \quad (\neg a \vee b \vee \neg c) \quad (\neg a \vee \neg b \vee c)$$

- Backtrack!

$$a = \mathrm{F} \quad \cancel{b = \mathrm{F}}$$
$$b = \mathrm{T}(\text{forced}) \quad \cancel{c = \mathrm{F}}$$
$$c = \mathrm{T}(\text{forced})$$

and obtain a conflict:



# Basic DLL V

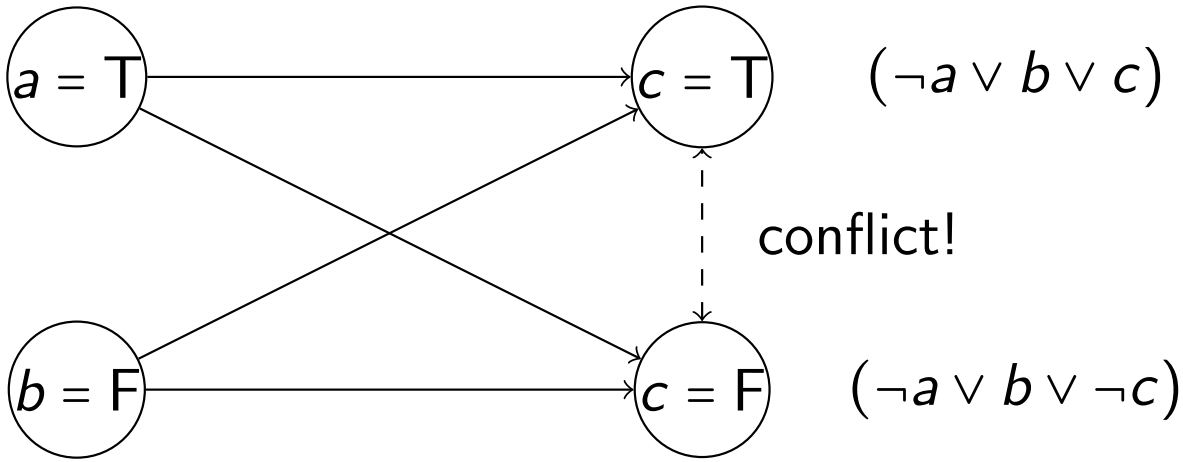
- Consider the following set of clauses:

$$(\neg a \vee b \vee c) \quad (a \vee c \vee d) \quad (a \vee c \vee \neg d) \quad (a \vee \neg c \vee d)$$
$$(a \vee \neg c \vee \neg d) \quad (\neg b \vee \neg c \vee d) \quad (\neg a \vee b \vee \neg c) \quad (\neg a \vee \neg b \vee c)$$

- Backtrack!

$$\cancel{a = \mathsf{F}}$$
$$a = \mathsf{T}(\text{forced}) \quad b = \mathsf{F}$$

and obtain a conflict:



# Basic DLL VI

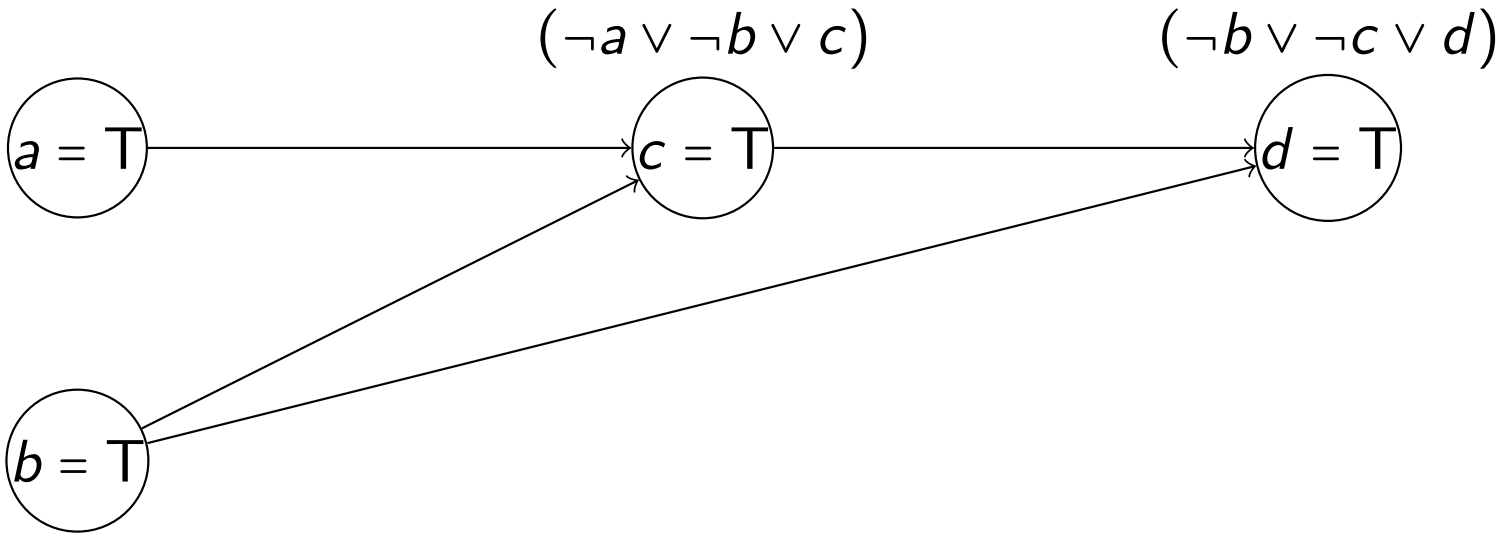
- Consider the following set of clauses:

$$(\neg a \vee b \vee c) \quad (a \vee c \vee d) \quad (a \vee c \vee \neg d) \quad (a \vee \neg c \vee d)$$
$$(a \vee \neg c \vee \neg d) \quad (\neg b \vee \neg c \vee d) \quad (\neg a \vee b \vee \neg c) \quad (\neg a \vee \neg b \vee c)$$

- Backtrack!

~~$a = \mathrm{F}$~~
$a = \mathrm{T}$(forced) ~~$b = \mathrm{F}$~~
$b = \mathrm{T}$(forced)

and SAT!



# Conflict Driven Learning

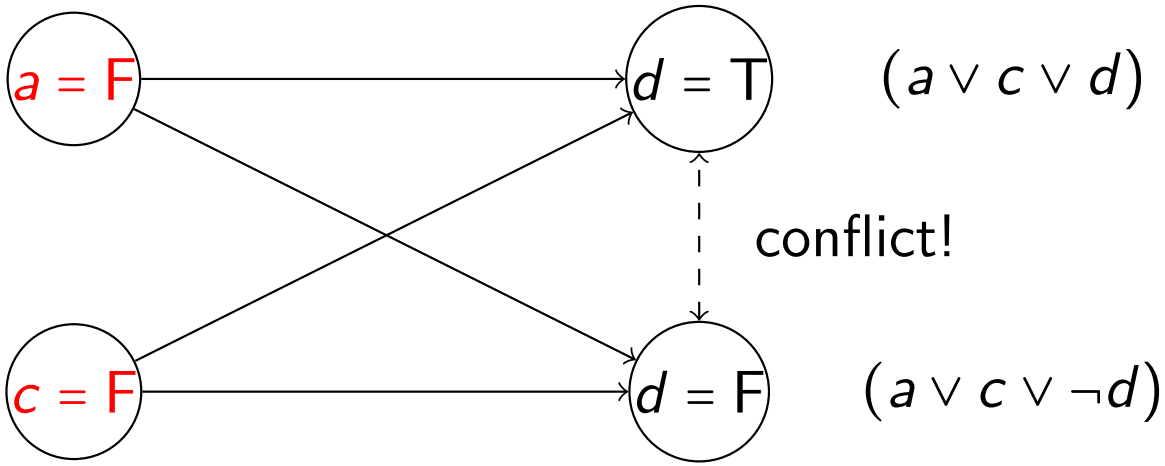
- When a conflict is encountered, add its cause to prevent it.
- Consider the following set of clauses:

$$(\neg a \vee b \vee c) \quad (a \vee c \vee d) \quad (a \vee c \vee \neg d) \quad (a \vee \neg c \vee d)$$
$$(a \vee \neg c \vee \neg d) \quad (\neg b \vee \neg c \vee d) \quad (\neg a \vee b \vee \neg c) \quad (\neg a \vee \neg b \vee c)$$

  and the following valuation:

$$a = \mathrm{F} \quad b = \mathrm{F} \quad c = \mathrm{F}$$

- The conflict is caused by $a = \mathrm{F}$ and $c = \mathrm{F}$:




- Hence we add a learned clause: $(a \vee c)$.

# Non-Chronological Backtracking

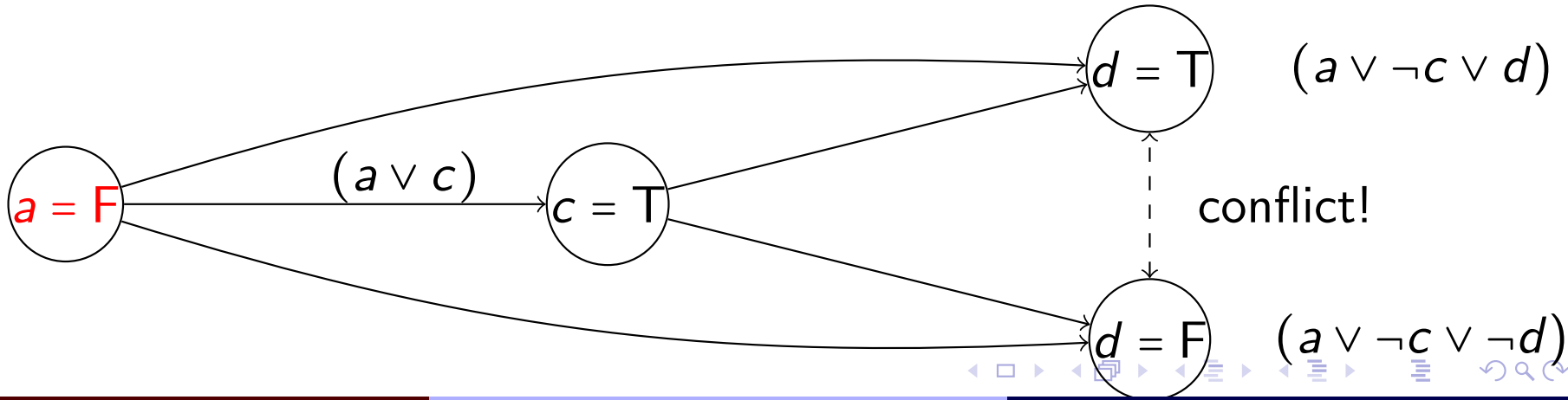
- When a learned clause is generated, backtrack to the next-to-the-last variable in the clause.
- Consider the following set of clauses:

$$(\neg a \vee b \vee c) \quad (a \vee c \vee d) \quad (a \vee c \vee \neg d) \quad (a \vee \neg c \vee d)$$
$$(a \vee \neg c \vee \neg d) \quad (\neg b \vee \neg c \vee d) \quad (\neg a \vee b \vee \neg c) \quad (\neg a \vee \neg b \vee c)$$
$$(a \vee c)$$

backtrack to $a$:

$$a = \mathrm{F}$$

obtain a conflict, and add a learned clause $(a)$:



# DPLL I

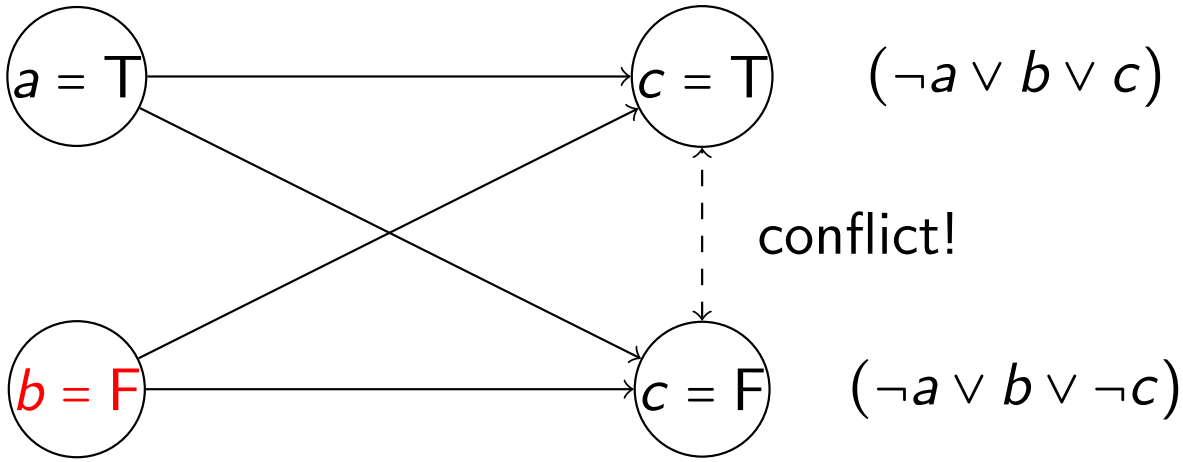
- Consider the following set of clauses:

$$(\neg a \vee b \vee c) \quad (a \vee c \vee d) \quad (a \vee c \vee \neg d) \quad (a \vee \neg c \vee d)$$
$$(a \vee \neg c \vee \neg d) \quad (\neg b \vee \neg c \vee d) \quad (\neg a \vee b \vee \neg c) \quad (\neg a \vee \neg b \vee c)$$
$$(a \vee c) \qquad (a)$$

- backtrack all variables:

$$b = \mathrm{F}$$

obtain a conflict, and add a learned clause $(b)$:

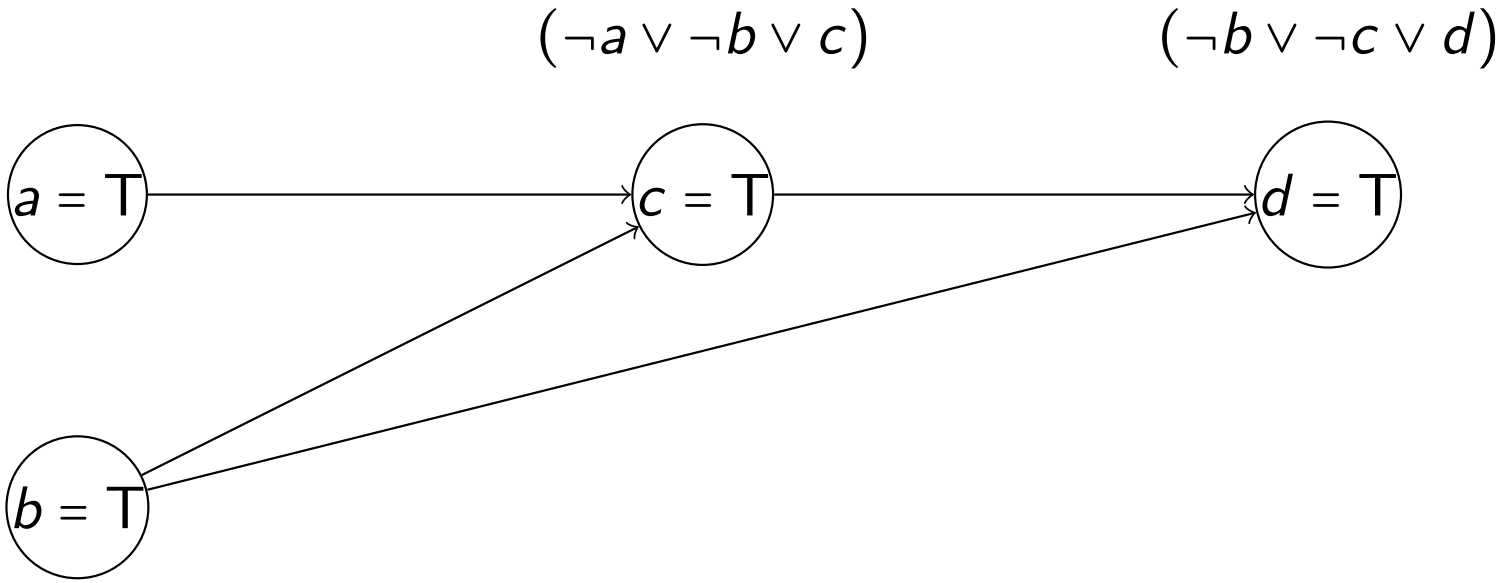


# DPLL II

- Consider the following set of clauses:

$$(\neg a \vee b \vee c) \quad (a \vee c \vee d) \quad (a \vee c \vee \neg d) \quad (a \vee \neg c \vee d)$$
$$(a \vee \neg c \vee \neg d) \quad (\neg b \vee \neg c \vee d) \quad (\neg a \vee b \vee \neg c) \quad (\neg a \vee \neg b \vee c)$$
$$(a \vee c) \qquad (a) \qquad (b)$$

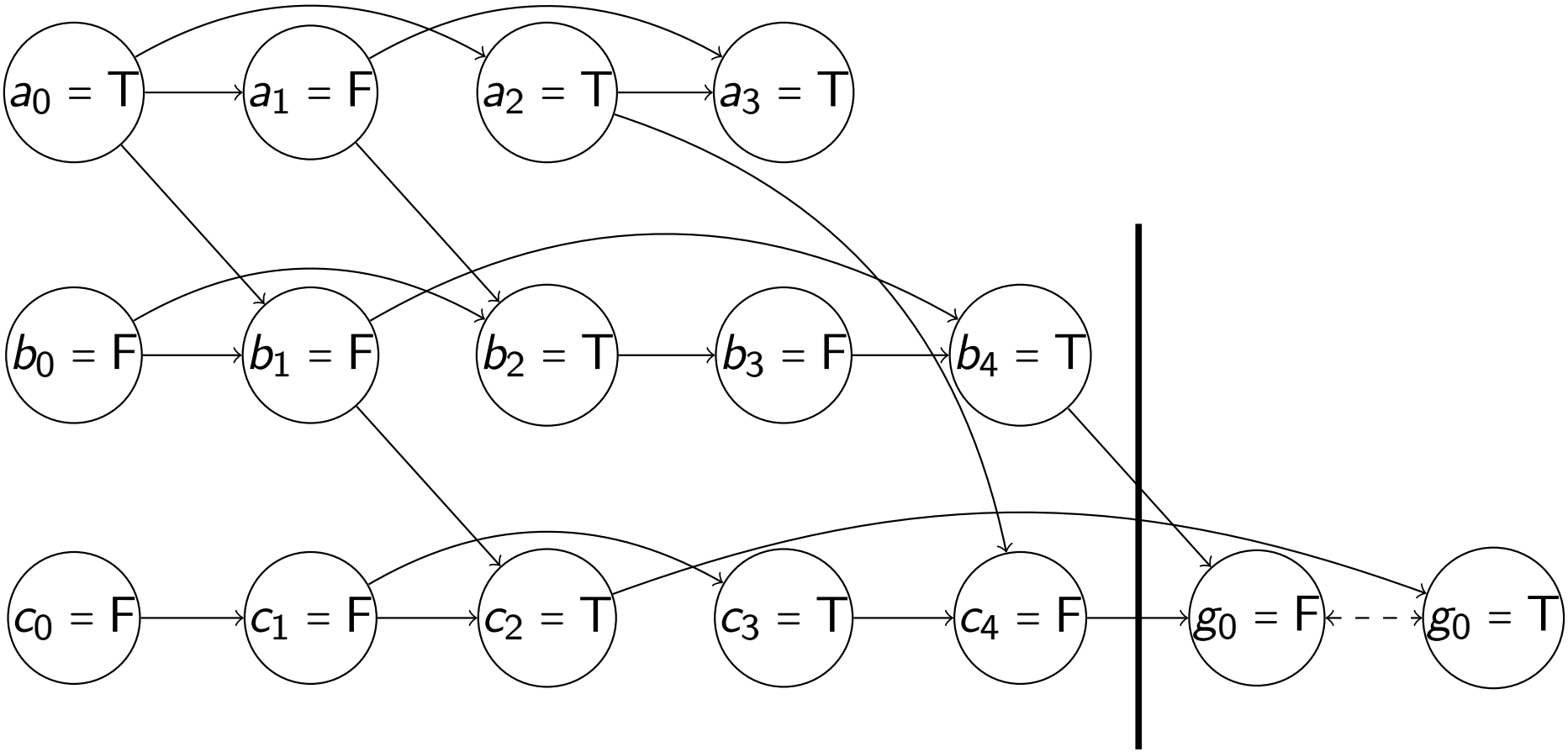
- and SAT!



# Conflict Analysis I

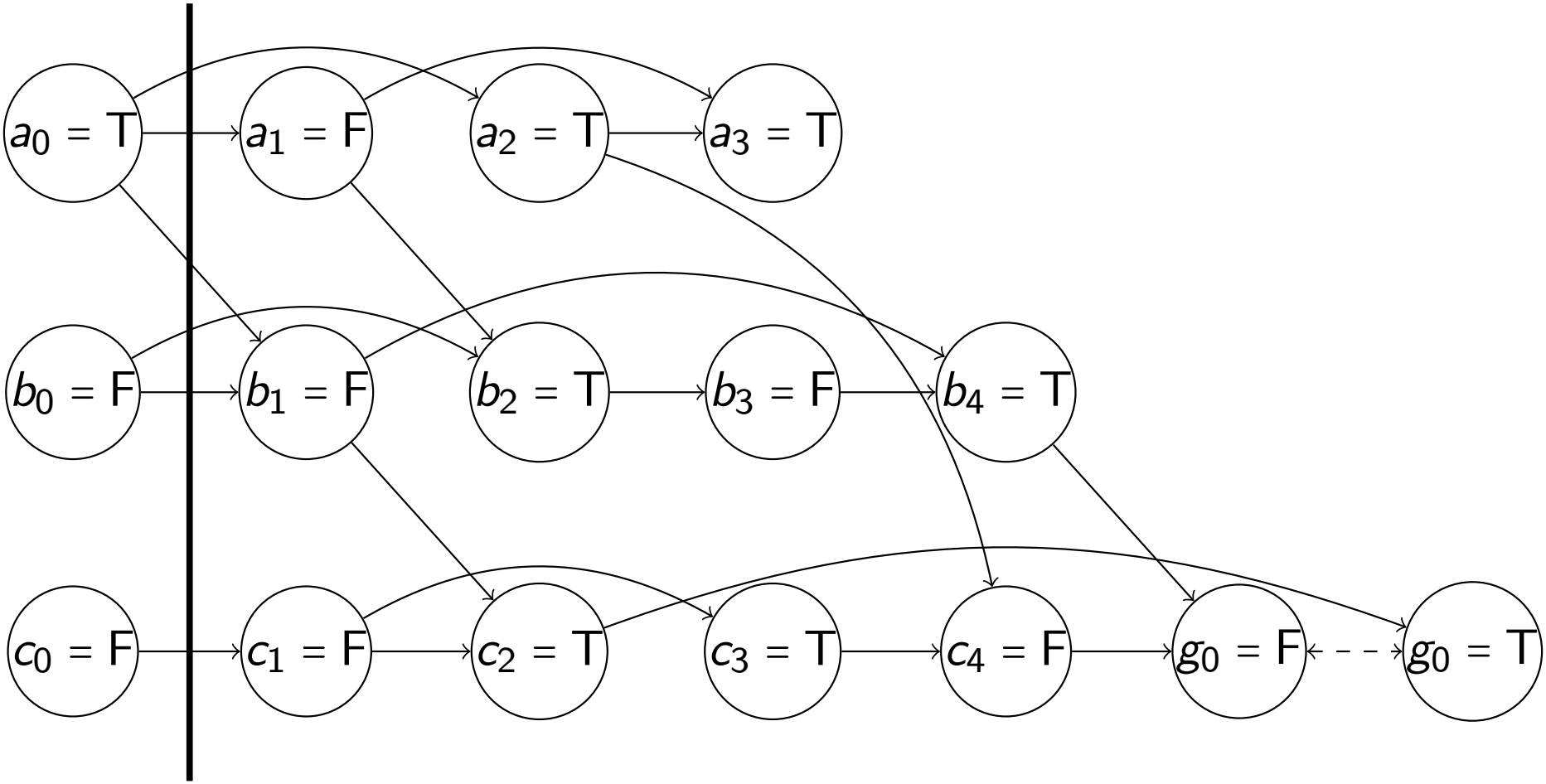
- When a conflict occurs, we would like to "learn" a clause to prevent the conflict from reoccurring.
- We hence would like to know what valuations cause the conflict.
- In an implication graph, any cut from root assignments to conflicting assignments is such a cause.

# Conflict Analysis II



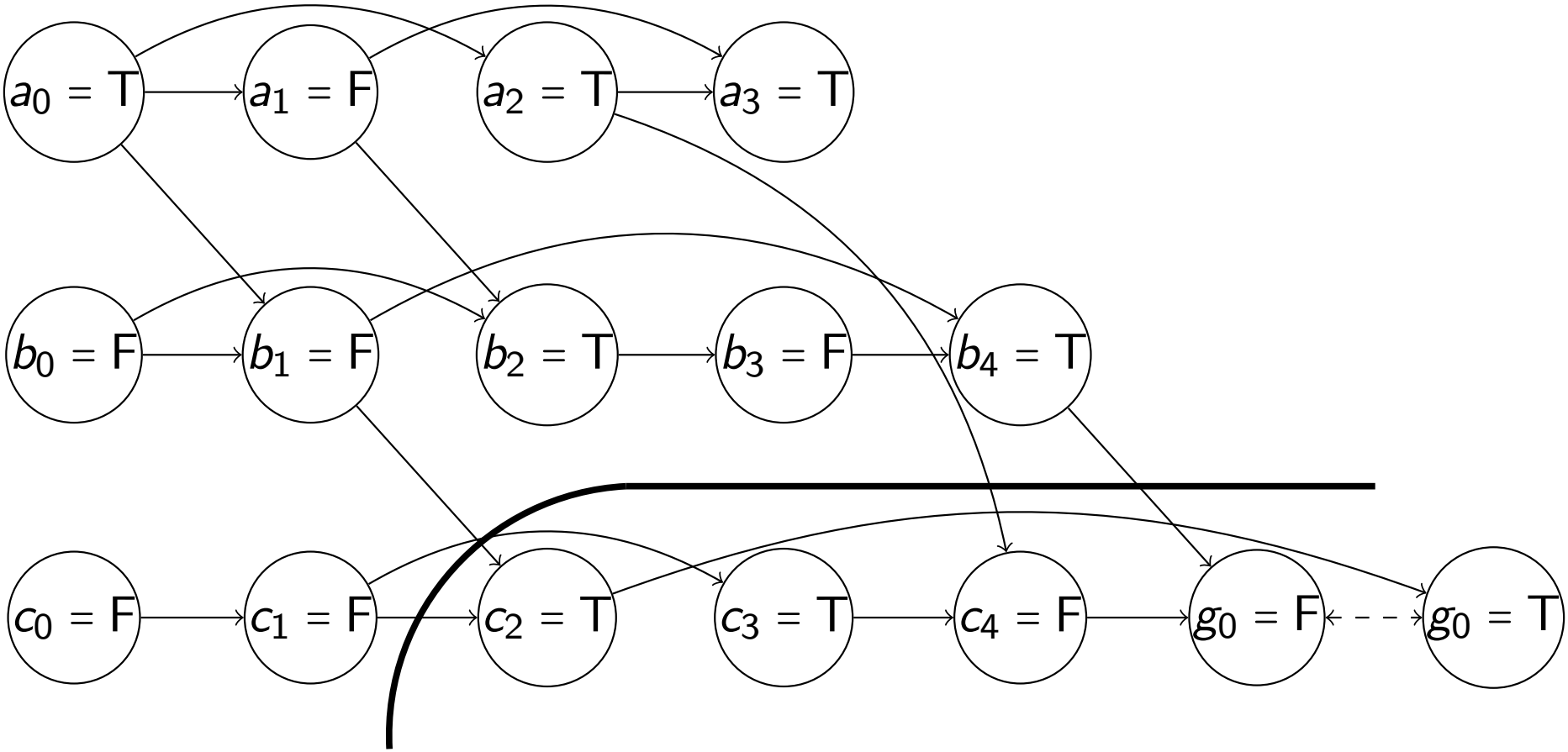

- $b_4 = \mathsf{T}$, $c_2 = \mathsf{T}$, and $c_4 = \mathsf{F}$ cause the conflict.
- Hence we can learn $(\neg b_4 \vee \neg c_2 \vee c_4)$.

# Conflict Analysis III




- $a_0 = \mathrm{T}$, $b_0 = \mathrm{F}$, and $c_0 = \mathrm{F}$ cause the conflict.
- Hence we can learn $(\neg a_0 \vee b_0 \vee c_0)$.

# Conflict Analysis IV

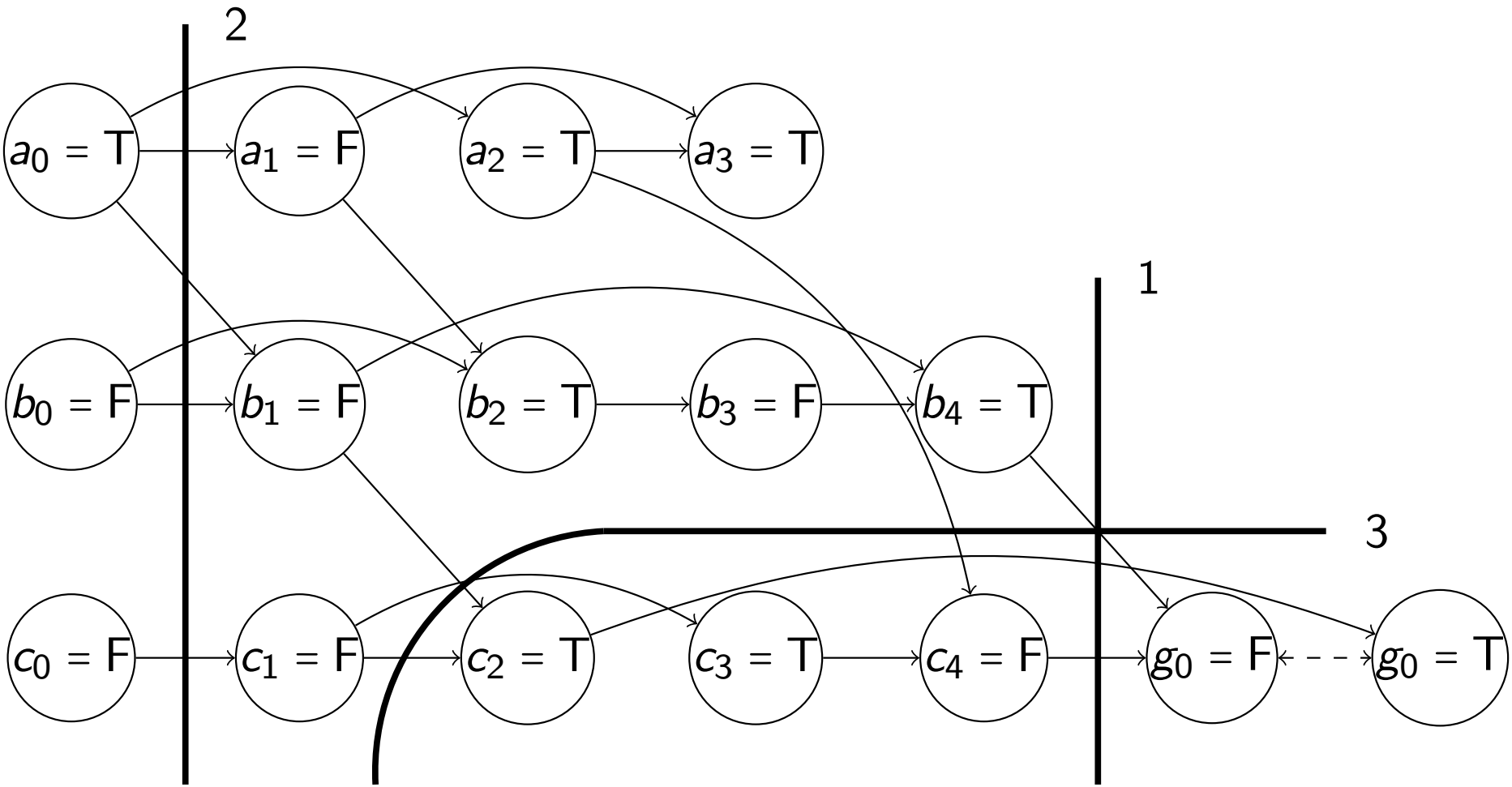



- $a_2 = \mathrm{T}$, $b_1 = \mathrm{F}$, $b_4 = \mathrm{T}$, and $c_1 = \mathrm{F}$ cause the conflict.
- Hence we can learn $(\neg a_2 \vee b_1 \vee \neg b_4 \vee c_1)$.

# Conflict Analysis V

- Clearly, there are several cuts in an implication graph.
- Each cut corresponds to a learned clause.
- Which one is the best?
- Idea: reverse exactly one truth value.
- Consider only cuts that has only one node in the same level as the conflict.
  - This is called a unique implication point (UIP).

# Conflict Analysis VI




- Cut 1 is not UIP. Cut 2 is the last UIP. Cut 3 is the first UIP.
  - Emperically, the first UIP is the best.

# Where to go?

- Contrary to general belief, SAT solvers are not impractical.
- Engineering ideas are essential to the real world.
  - When cleverly applied, they can even tackle hard theoretical problems such as satisfiability.
- We only discuss a clever idea in SAT solvers very briefly.
- There are certainly many more.
  - There are many interesting ideas for propagation, memory management, etc.
- MiniSAT is an open-sourced fast SAT solver.
- Read its code. You will learn a lot more!